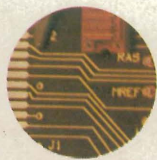
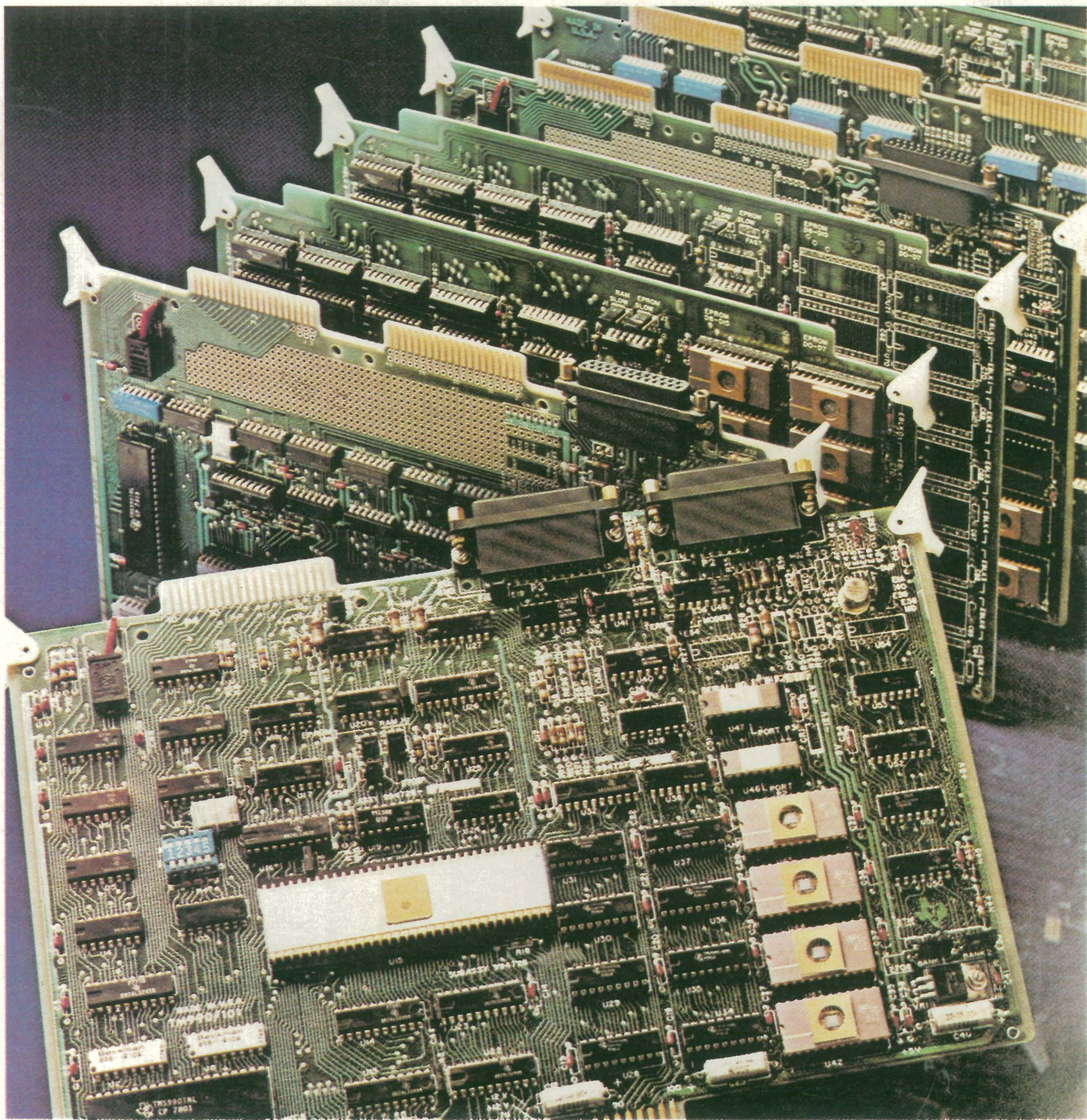
UNA PUBBLICAZIONE
DEL GRUPPO EDITORIALE JACKSON
ANNO 2 N. 4
Bit
MICROPROCESSORS-HARDWARE - SOFTWARE
HOME & PERSONAL COMPUTERS
L. 2000
Spedizione in abb. postale Gruppo IV/70

La Texas Instruments annuncia il primo modulo di microcomputer a 16 BIT con linguaggio BASIC. Più facile da programmare, ampliato nella capacità di memorizzazione e di comunicazione

Sono disponibili presso i distributori autorizzati T.I., dei veri e propri computers su singola scheda.
Stiamo parlando dei nuovi moduli di microcomputers TM 990/101M.
La maggiore estensione di memoria, la possibilità di comunicazioni simultanee su due porte (con protocolli di comunicazione già implementati su board), e la disponibilità del POWER BASIC, linguaggio ad alto livello che rende estremamente veloce la programmazione, sono la caratteristiche fondamentali di questi moduli.

Questi nuovi moduli della Texas Instruments riducono i tempi di progetto e di sviluppo, il numero di

componenti del sistema ed i costi di realizzazione.

Queste unità sono preassemblate, pretestate e pronte all'uso, e questo garantisce un'alta affidabilità al sistema.

Estensione di memoria quattro volte più grande

I microcomputers TM 990/101M possiedono una capacità di memoria RAM statica su board che va da 1K a 2K parole da 16 bits.
L'estensione di EPROM va da 2K a 4K parole da 16 bits.

Capacità di comunicazione doppia

Sul board sono disponibili due porte per comunicazioni seriali: una usata per utilizzazioni "remote" (terminali o modem), l'altra per utilizzazioni "locali" (terminali EIA, teletype, microterminale TM 990/301 della T.I.).

Il meglio nell'ambito dei microcomputers

Nella serie TM 990 della T.I. troverete la più ampia scelta possibile per prestazioni/costo, di microcomputers a 16 bits che possono soddisfare le vostre necessità a livello di sistema. Ideali per valutazioni su microprocessori e per accelerare il progetto di sistemi a microprocessori, rappresentano una valida alternativa in produzione.
Caratteristica fondamentale di tutti i moduli è la completa compatibilità a livello di set di istruzioni con gli altri membri dell'intera famiglia TM 990/9900.

Per valutazioni ed applicazioni OEM

- TM 990/100M - Utilizza il microprocessore TMS 9900 a 16 bit in tecnologia N MOS, della Texas Instruments.

  1K bytes di RAM statica, 2K bytes di EPROM, e porte di I/O seriali e parallele programmabili, fanno di questa unità un potente microcomputer su singola scheda.

- TM 990/180M - Utilizza il microprocessore TMS 9980 a 16 bit, in tecnologia N MOS, della T.I., che consente di lavorare a 2,5 MHz, e che contengono una interfaccia a 8 bit per la memoria, minimizza i costi dei sistemi più piccoli.

Per espansioni di memoria

- TM 990/201 - 8K bytes di EPROM e 4K bytes di RAM statica. Espandibile a 32K bytes di EPROM e 16K bytes di RAM.
- TM 990/206 - 8K bytes di RAM statica espandibile a 16K bytes.

Per ingresso dati e monitoraggio

- TM 990/301 - Consente l'ingresso di dati di programma, la visualizzazione e la modifica del contenuto dei registri interni e della memoria sotto controllo software (TIBUG).

Per espandere le linee di I/O

- TM 990/310 - Modulo di espansione delle linee di I/O a 48 bit.

Per sviluppare software e per la produzione

- TM 990/401 - Monitor interattivo di debug (TIBUG) già programmato in EPROM.
- TM 990/402 - Assemblatore line-by-line già programmato in EPROM.
- TM 990/450 - POWER BASIC già programmato in 8K bytes di E-PROM.
- TM 990/451 - POWER BASIC già programmato in 12K bytes di E-PROM.

Sono inoltre disponibili una vasta gamma di connettori e cavi per gli usi più comuni, schede di prototipizzazione, ed uno chassis a 4-slot (TM 990/510) già predisposto per ospitare 4 schede della famiglia TM 990.

Supporto software

I moduli di microcomputer della serie TM 990 sono completamente supportati dal sistema di sviluppo AMPL (Advanced Microprocessor Prototyping Laboratory) della Texas Instruments.

L'AMPL consente di effettuare trace a 10 MHz ed emulazioni per i microprocessori TMS 9900, 9980, SBP 9900, TMS 9940 e per tutti i microprocessori della T.I. che saranno prodotti in futuro.

L'AMPL è disponibile nella configurazione con sistema a floppy disk o in quella con sistema a dischi rigidi in modo da soddisfare le più svariate esigenze degli utenti. I programmi possono essere editati, assemblati, lincati, caricati ed eseguiti più velocemente che con i sistemi tradizionali a nastri o a cassette.

L'emulazione del TMS 9900/9980 consente di sviluppare e di debuggare il software direttamente su un modulo TM 990 visualizzando e controllando, contemporaneamente, le operazioni per mezzo del sistema di sviluppo AMPL.

Per salvaguardare il progetto di oggi dalle innovazioni di domani

I microcomputers della serie TM 990 e l'AMPL si integrano perfettamente con l'intera famiglia 990/9900 rendendola veramente completa e di estema potenzialità nell'ambito delle applicazioni a 16 bit.

Questa famiglia è composta da una vasta gamma di microprocessori, microcomputers e minicomputers che usano tutti la stessa architettura orientata verso la memoria, lo stesso set di istruzioni; sono tutti completamente compatibili e sono previsti per essere supportati dallo stesso sistema di sviluppo.

Con la famiglia 990/9900 il progettista può scegliere la soluzione a costo più basso, e più soddisfacente alle sue esigenze, senza preoccuparsi degli sviluppi futuri del suo prodotto, in quanto la completa compatibilità software rende il progetto software direttamente utilizzabile su prodotti che si possono diversificare per sofisticazione hardware.
Per maggiori informazioni contattate i nostri distributori autorizzati oppure l'ufficio Promozione Commerciale, Cittaducale - RIETI

MICROCOMPUTER SU SCHEDA SINGOLA

AIM 65

AIM 65:
il microcomputer che ha nella sua grande versatilità d'impiego il suo maggior pregio: sistema di sviluppo, controllo di processo, tester, terminale, sistema di istruzione... e poi basta solo un po' di fantasia per trovare altre mille utili applicazioni.
Anche il prezzo è quanto mai interessante!
L'AIM 65 è completo di: stampante caratteri ASCII 20 colonne – display 20 caratteri ASCII – interfaccia per due audio cassette e TTY – tastiera completa di tipo terminale – 1 K o 4 K byte RAM – bus espandibile esternamente.
Firmware: – monitor – debugger (trace, break points) – assembler – disassembler – text editor – basic.

Dott. Ing. Giuseppe De Mico s.p.a.

20121 MILANO
Via Manzoni, 31
Tel. (02) 653131 - Telex: 312035
Telegr.: Twinrapid
Uffici regionali:
Roma/Torino/Bologna/Padova

SOMMARIO

Vol. 4 N° 4 - Settembre/Ottobre 1979

In copertina:
Il successo di BIT '79
Grafica di R. Nisi

INDICE INSERZIONISTI

DIRETTORE RESPONSABILE
Giampietro Zanga

COORDINATORE TECNICO
Marcello Montedoro

CAPO REDATTORE
Dino Bortolossi

SEGRETARIA DI REDAZIONE
Cecilia De Serio

GRAFICA E IMPAGINAZIONE
Job Line srl

CONSULENZA E COORDINAMENTO
Marcello Montedoro

DIFFUSIONE E ABBONAMENTI
Gabriella Napoli, Silvia Decari
Marco Benedetti

DIREZIONE, REDAZIONE
P.le Massari, 22 - 20125 Milano
Telefoni 68.03.68 - 68.00.54

AMMINISTRAZIONE
Via Vincenzo Monti, 15 - 20123 Milano

PUBBLICITA': Concessionario per l'Italia e l'Estero Reina & C. S.n.c. - Via Ricasoli, 2 - 20121 Milano
Tel. (02) 803101-866192
Via Saverio Carmignano, 10 - 00151 Roma Tel. (06) 5310351

FOTOCOMPOSIZIONE: New Comp
Via S. Michele al Carso, 5 - Nova Milanese

STAMPA: Litografia del Sole srl
Buccinasco

Concessionario esclusivo per la diffusione in Italia e all'Estero: SODIP - Via Zuretti, 25 - 20125 Milano

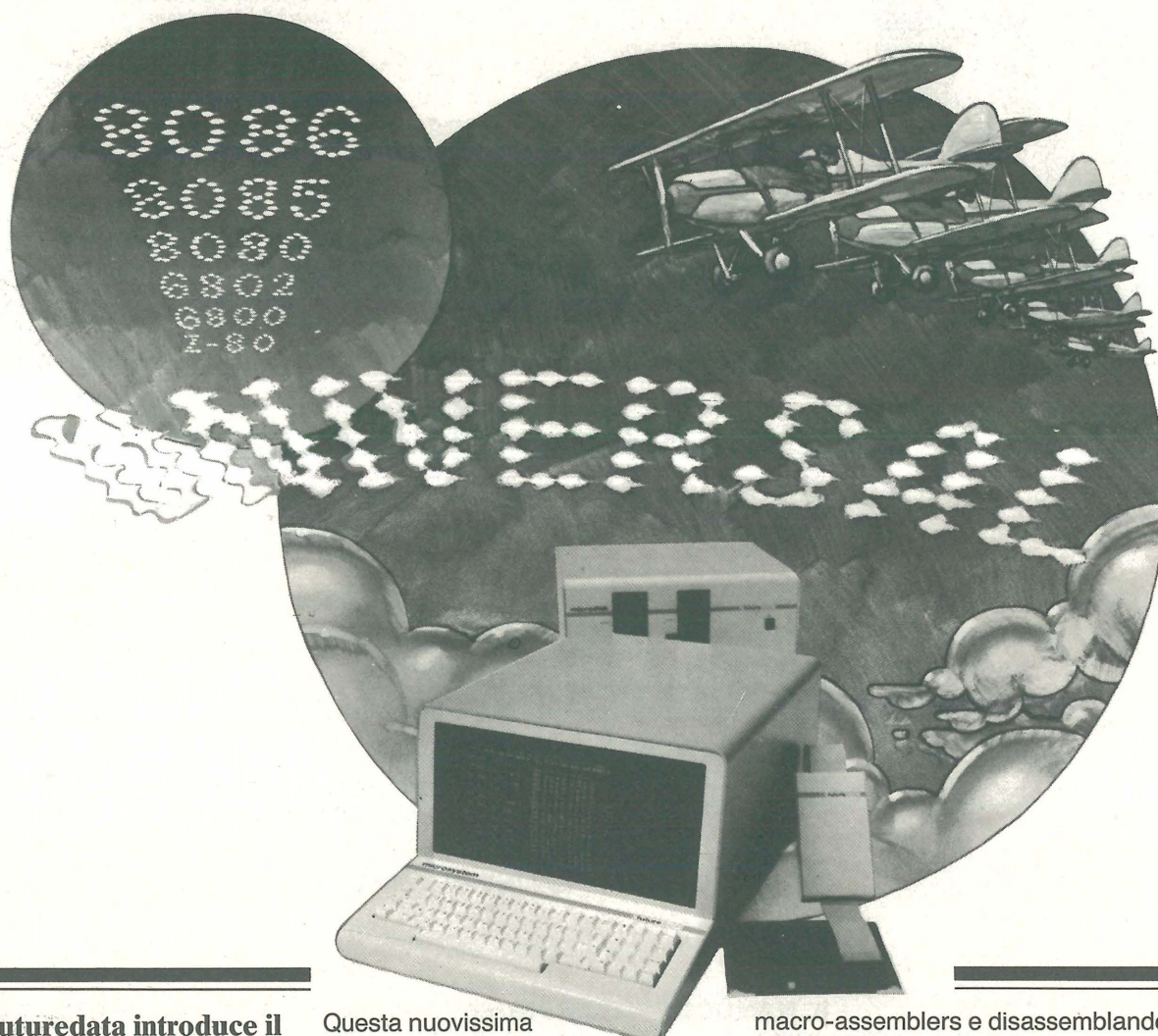
Spedizione in abbonamento Postale Gruppo IV/70

Prezzo della rivista L. 2.000

Numero arretrato L. 3.000
Abbonamento annuo L. 8.000
per l'Estero L. 12.000

I versamenti vanno indirizzati a: Jackson Italiana Editrice S.r.l. Via V. Monti, 15 - 20123 Milano mediante emissione di assegno bancario, cartolina vaglia o utilizzando il c/c Postale numero 11666203
Per i cambi d'indirizzo, indicare, oltre naturalmente al nuovo, anche l'indirizzo precedente, ed allegare alla comunicazione l'importo di L. 500, anche in francobolli.

Advanced Microcomputer Development System

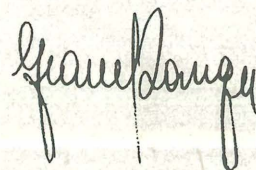
µPµP and away

Futuredata introduce il sistema di sviluppo per microprocessori che vi rende liberi.

Questa nuovissima realizzazione nel campo dei sistemi di sviluppo per microprocessori vi permette di far fronte alla rapida evoluzione di questo settore.
Sarete liberi di progettare con l'8086, 8080, 6800, 6802 o Z-80 e liberi di aggiungere, presto, molti altri processori.
Questo sistema vi mette a disposizione possibilità di sviluppo di hardware e software: emulazione in-circuit fino a 5 MHz in tempo reale, analisi logica a 48 canali in tempo reale, 2 Megabytes di memoria su disco, e ogni supporto di software, incluso compilatori di linguaggio ad alto livello, riallocando macro-assemblers e disassemblando messe a punto simboliche.
CPU, video e tastiera sono tutti nettamente integrati in contenitore trasportabile per lasciare più spazio libero sul vostro tavolo.
È l'universale, completo, ultimo ritrovato della tecnologia AMDS: Advanced Microcomputer Development System.

DIVISIONE DELLA

GenRad

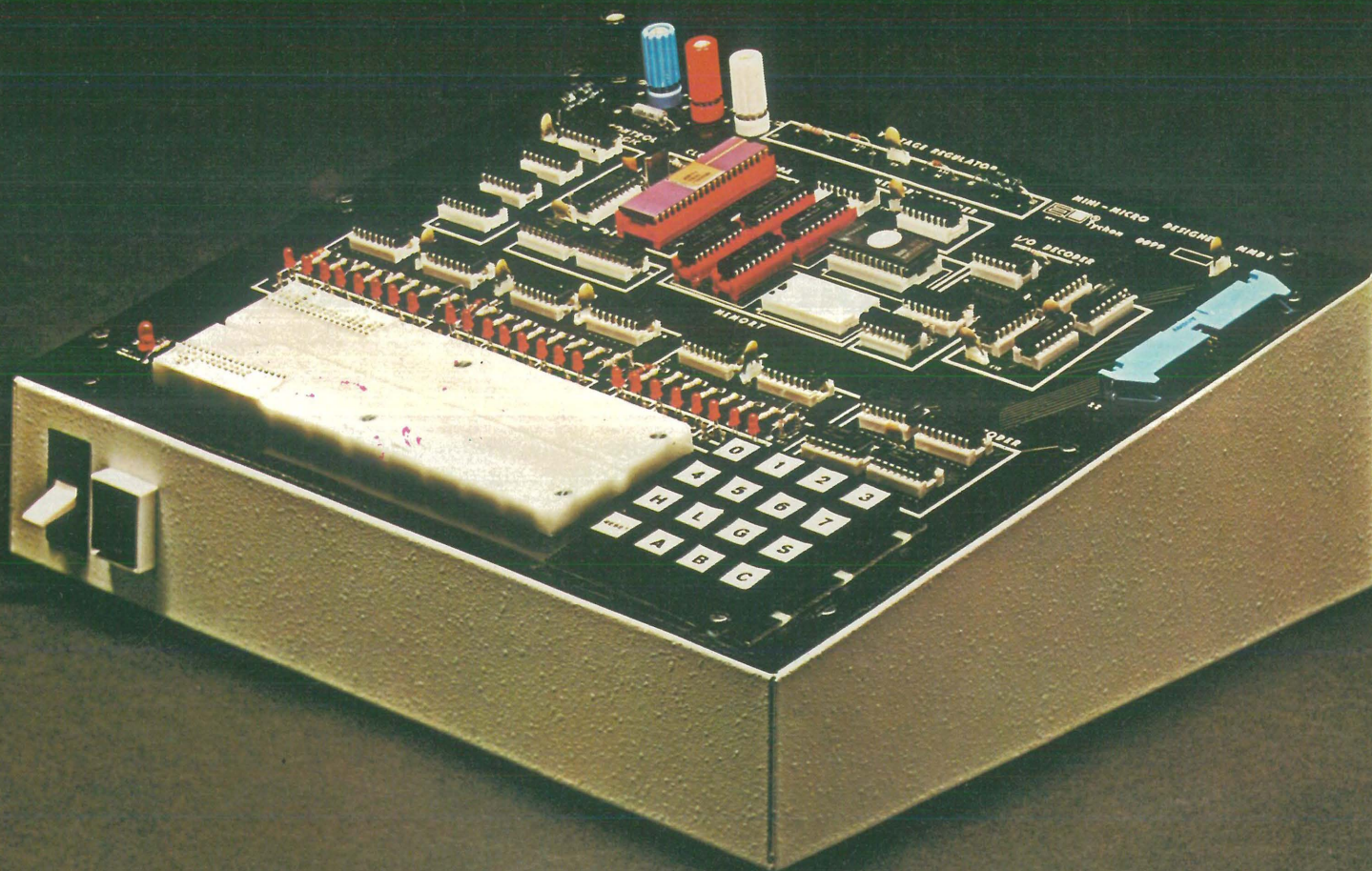
GenRad S.p.A. 20141 MILANO
Via Lampedusa, 13
tel. 84.66.541 telex 320373

EDITORIALE

Non leggete qui sotto: è un ordine!

È già il quarto numero di BIT; è tempo di tirare le somme per vedere se e come continuare. Non che abbiamo mai preso sul serio, in Jackson, l'idea di smettere davvero: è solo un modo di dire, una protesta di umiltà e di disposizione all'autocritica che nella sua ipocrisia non convince nessuno. Ma almeno, anche se per scherzo, il problema ce lo poniamo costantemente, perché i lettori sono la parte più dannatamente importante di una rivista.

Così abbiamo bisogno del vostro parere. Dovete dirci se andiamo bene così, o se e dove dobbiamo correggere il tiro. Vogliamo sapere da voi quali sono gli argomenti che apprezzate di più, quali vi interessano di meno, quali altri argomenti vorreste veder trattati, e come. E questo non è uno scherzo: ce lo aspettiamo davvero. L'indirizzo a cui scrivere l'avete, carta e penna non vi mancano certo, né vi mancano le idee (altrimenti non sapreste che farvene di questa rivista). Se supererete la naturale pigrizia che di regola impedisce di congegnare questi elementi in un parere scritto e coerente, vi ritroverete con una rivista migliore e meglio attagliata a voi. Oltre a tutto, ci farete un piacere enorme: ad uno dei nostri collaboratori un paio di giorni fa qualcuno di voi ha rivolto un complimento per BIT; c'è voluto del bello e del buono per farlo smettere di pavoneggiarsi allo specchio.

Sì, siamo tutti un po' narcisisti, ci piace rileggerci una volta stampati e ci piace che ci si dica che abbiamo fatto un buon lavoro; ma, per carità, non venite a dirci solo questo. Tirate fuori le critiche che certo avete: fanno un po' male all'inizio, ma permettono di incassare molti più complimenti poi.

A proposito del titolo qui sopra: complimenti per aver disubbidito. Molte persone si lasciano intimidire e influenzare dalla carta stampata in una maniera decisamente indecente ("...è proprio vero: c'è scritto anche nel giornale!"); così invece di adattare il giornale o la rivista ai lettori, sono questi ultimi che si adattano. Siamo lieti che non apparteniate a questo squallido gregge; permetteteci di approfittarne, dite la vostra.

Gianni Longo

SINTESI

Small business computers: alcune indicazioni

La comparsa, sul mercato, degli small business computers, ovvero di sistemi per applicazioni gestionali realizzati con microcomputers, allarga il campo dell'E.D.P. a settori per i quali il fattore investimento iniziale era e rimane il più grosso ostacolo anche se non il solo. In quale misura questo ostacolo è stato rimosso dalla introduzione di sistemi a microelaboratore è ancora presto per dirlo, quantunque le iniziative commerciali verso tali settori siano ormai più numerose e più aggressive, giusto per superare una naturale impreparazione di questa nuova utenza nei confronti dell'elaborazione elettronica ed una altrettanto naturale anche se immotivata diffidenza nei confronti di marchi sconosciuti ai più.

Per valutare in quali termini si pone oggi la questione dell'investimento iniziale per un sistema a microcomputer per applicazioni gestionali, facciamo riferimento ad un hardware costituito dal computer vero e proprio, terminale video, tastiera, stampante e memoria di massa.

Questa configurazione è quella minima indispensabile: praticamente tutte le applicazioni gestionali richiedono come configurazione base lo stesso tipo di hardware, con la conseguenza che nella maggior parte dei casi esso è sottoutilizzato. (Il rovescio della medaglia è che l'eventuale esigenza di aumento di prestazione non richiederà sostanziali investimenti).

In dettaglio il costo delle singole parti è il seguente: computer vero e proprio (comprendente alimentatore, schede di I/O, almeno 48 Kbytes di memoria per contenere i vari programmi di elaborazione) attorno ai 3.500.000 di lire; stampante (fornita di introduttori frontali per l'inserimento di moduli prestampati e con possibilità di stampa anche sul giornale di cassa, etc.) attorno ai 2.000.000 di lire; terminale video (24 linee di 80 caratteri) e tastiera alfanumerica (fornita anche di tastiera numerica) attorno a 1.500.000 di lire; infine, scegliendo come memoria di massa la soluzione che vede due floppy-disk con una capacità di almeno 500.000 Kbytes, il costo ulteriore da considerare si aggira attorno ai 3.000.000.

In conclusione il costo totale è dell'ordine dei 10.000.000 di lire: questo è dunque il costo minimo dell'hardware di un sistema per applicazioni gestionali.

Si è detto che in tale ambito il sistema è sottoutilizzato: un tipico esempio è la stampante, che non viene impiegata per tutto il tempo di utilizzo del sistema, e d'altra parte prenderne in considerazione altre più economiche non migliora di molto il risultato, ed in prospettiva è una scelta negativa. Anche il computer vero e proprio è sottoutilizzato, in quanto è in grado di sopportare un carico di lavoro ben maggiore di quello richiesto mediamente in questo tipo di applicazioni, senza degradamento delle prestazioni.

La sola limitazione che esiste è quella relativa alla capacità della memoria di massa, ma in questo caso si può ovviare ricorrendo alla sostituzione manuale dei dischetti opportuni, posto che non vi siano vincoli di avere in linea un numero superiore di Kbytes.

Il costo indicato non è quello finale, perché non comprende il costo del software applicativo (che merita un discorso a parte): è da precisare però che il software di base è solitamente fornito insieme all'hardware.

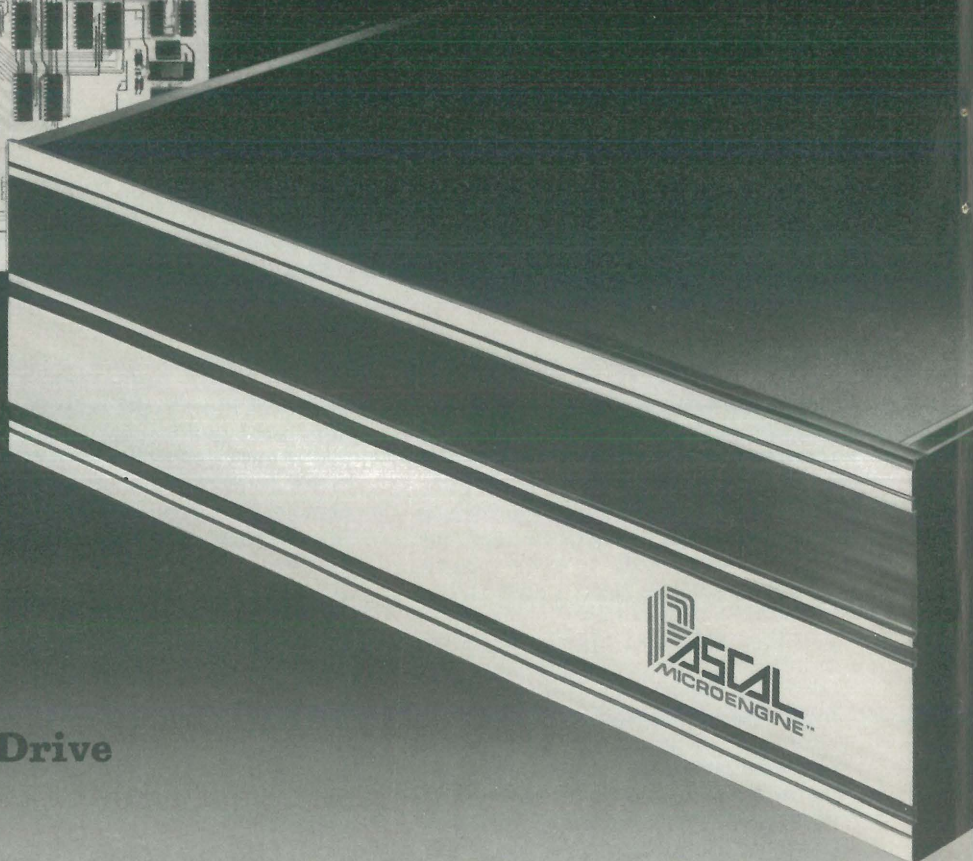
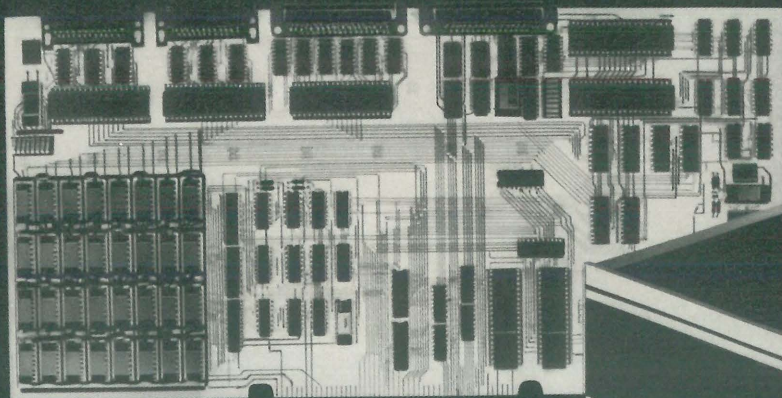
Ad ogni buon conto quello che merita di essere sottolineato è che la cifra precedentemente indicata è il valore minimo, al di sotto del quale non si può correttamente parlare di sistemi per applicazioni gestionali, ed è presumibile che questa situazione permanga per un anno almeno. Infatti il costo del computer vero e proprio ha ormai raggiunto un limite al di sotto del quale non si può scendere significativamente: distinguendo tra C.P.U. e memoria, a quest'ultima soltanto è possibile attribuire la capacità di diminuire il prezzo del computer. Ma d'altra parte, perché questa diminuzione sia significativa, occorre affidarsi ad ulteriori sviluppi tecnologici che a tutt'oggi si chiamano memorie a bolle e memorie CCD, non ancora attuali e competitive.

Per quanto concerne i sistemi periferici, il discorso non è migliore, primo perché sono questi che incidono maggiormente sul costo globale del sistema, secondo perché utilizzano componenti meccanici, elettromeccanici e di elettronica tradizionale che sono soggetti ad inflazione e non beneficiano di quei progressi tecnologici che hanno ridotto notevolmente il prezzo, ad esempio, della C.P.U. e delle memorie.

In conclusione, il costo di un microcomputer per applicazioni gestionali è attestato su valori che realisticamente, nel breve periodo, non saranno soggetti a diminuzioni significative, per cui occorre considerare attentamente questa indicazione all'interno di un discorso di investimenti non dilazionabili.

MARCS

WD/90
Pascal MICROENGINE™

PASCAL MICROENGINE™

- 64 K Byte RAM
- Floppy Controller per 4 Drive
  singola/doppia faccia
  singola/doppia densità
- 2 Interfacce seriali
- 2 Interfacce parallele
- Usa lo stesso "Chip-set" dell' LSI 11*
- Microprogrammato per eseguire direttamente istruzioni in PASCAL
- Prezzo 10 up:
  sistema completo di software: 2.950.000 Lit. più IVA
  scheda logica: 1.950.000 Lit. più IVA

*LSI è un marchio registrato DEC

COMPREL s.r.l.

20092 CINISELLO B. (MI) - VIALE ROMAGNA, 1
☎ (02) 6120641/2/3/4/5 - Telex: 332484 COMPRL I

Uffici regionali:
40137 BOLOGNA - P.za Azzarita, 6 - Tel. (051) 551306
50127 FIRENZE - Via T. Mabellini, 4 - Tel. (055) 412018
16033 LAVAGNA (GE) - P.za Marini, 20/10 - Tel. (0185) 301100
60025 LORETO (AN) - Via Dante Alighieri, 26/B - Tel. (071) 977693
35100 PADOVA - Via R. De Visiani, 17 - Tel. (049) 750741
00141 ROMA - Via Muzio Clementi, 58/5 - Tel. (06) 3603463 - 3600971
10144 TORINO - Via G. Fagnano, 10 - Tel. (011) 472789

NEWSLETTER

La RCA annuncia Il VIP-II a 400 $

Anticipando quella che sarà la tendenza futura del mercato, anche la RCA ha annunciato la sua strategia per l'home computer. Punto focale di essa sarà il VIP-II, un sistema basato sul microprocessore RCA 1802 e formato da 8 kilobits di RAM, 12 kilobits di ROM, una tastiera ASCII e con capacità sonore e di riproduzione a colori. La sua introduzione è però subordinata all'approvazione della FCC-Federal Communication Commission, dato che richiede un dispositivo di interfacciamento col monitor o il TVC. Il suo costo non supererà i 400 dollari.

La felice riscoperta del Pascal

Durante gli ultimi anni il costo di una unità di base di software, una singola linea di codice, è più che raddoppiato con tendenza ad un ulteriore incremento del 100% entro il 1985.Forse però questa volta non si arriverà a tanto. L'industria del software ritiene di aver trovato una soluzione all'ascesa dei costi. Più che di una scoperta si tratta di una riscoperta. Il suo nome è linguaggio Pascal. Con l'adozione di questo linguaggio al posto di quelli ora dominanti il mercato (Fortran, Basic e Cobol) è lecito aspettarsi riduzioni nei costi di programmazione attorno al 70% ed una maggiore produttività degli addetti alla programmazione. Sono in molti a ritenere che il Pascal, principalmente per la sua facilità di uso, diventerà il linguaggio più diffuso per i microprocessori. Alcuni costruttori lo hanno già adottato ad occhi chiusi. Si possono citare la Texas Instruments (l'80% di tutte le spese di ricerca e sviluppo per il software dei suoi microprocessori riguarda il Pascal), la General Automation (dirotta verso il Pascal il 100% dei suoi investimenti di software), la National Semiconductor (per tutti i nuovi microprocessori ed anche sistemi più potenti il Pascal è obbligatorio), la Motorola ed altri. Manca la Intel. Supporterà il Pascal, ma per i suoi nuovi microprocessori da 16 bit essa ha ritenuto preferibile sviluppare un suo proprio linguaggio, denominato PLM86. Anche i fornitori di medi-grossi sistemi pensano al Pascal. In attesa che il processo evolva parecchi hanno iniziato a proporre compilatori Pascal. A frenare la rincorsa di questo linguaggio provvedono principalmente due elementi: la mancanza di una standardizzazione e la necessità di non buttare a mare grossi investimenti compiuti in altri tipi di linguaggi di programmazione.

Meno ITT 20-20 del previsto

Il mercato europeo dei personal computer è in forte crescita. La ITT Europe non è tuttavia riuscita a vendere quante unità prevedeva. La colpa non è del mercato finale ma del mercato componentistico. La ITT afferma infatti di aver avuto difficoltà nel trovare i componenti necessari per realizzare i sistemi Apple che essa progetta e costruisce sulla base di un accordo con la omonima casa. Il sistema si chiama ITT 20-20. Nel nostro continente questi personal computer vengono costruiti con moduli importati dalla Apple, ma una sostenuta domanda in America dei sistemi di questo costruttore ha impedito di soddisfare i bisogni della ITT Europe nei tempi desiderati. Anche la penuria di alcuni semiconduttori chiave e di floppy-disk drives ha contribuito a far slittare le consegne.

Ormai sono in molti a dirlo:

MMD1

L'unico microcomputer didattico che lavora con entrambi i microprocessori

MMD1-A, assemblato £ 445.000+IVA
MMD1-K, in kit istruzioni in italiano £ 315.000+IVA

8080 A e Z-80*...

* con l'adattatore MMD1 - Z80

La Mattel si estende all'home computer

La strategia risponde ad una logica dello step-by-step ma in tempi brevi. Così la Mattel conta di approdare sul mercato dell'home computer dopo l'azzeccato approccio al mercato dei videogames. Per dimezzare i rischi e garantirsi nel contempo un valido fornitore di componenti elettronici, la Mattel nello scorso giugno si è accordata con la General Instrument per una collaborazione in questa direzione. Insieme queste due case hanno ideato un servizio (denominato Play Cable) per fare arrivare nelle abitazioni, via cavo, un complesso di programmi educativi, informativi e diversivi per i quali utilizzare un home terminal come punto di arrivo. Si tratta di un apparecchio, sviluppato e costruito dalla Mattel con componenti acquistati dalla Jerrold Electronics (consociata della General Instrument), offerto a 250 $. Questo terminale non è che un elemento di congiunzione per arrivare al calcolatore. La Mattel ha già pronto infatti il suo home computer che proporrà agli utenti del servizio Play Cable allo stesso prezzo di 250$ circa. In entrambi i casi è un microprocessor che fa da cuore ai sistemi. Di sviluppo e produzione Mattel saranno anche i programmi diffusi via rete di televisione via cavo oppure offerti sotto forma di cassetta. Incompatibilità con l'attività videogames? In Mattel ritengono che i due approcci possono coesistere a causa dei differenti requisiti che gli utenti hanno per i vari programmi.

La Video Brain ricomincerà daccapo?

Una delle prime vittime della battaglia scatenatasi sul mercato dei calcolatori personali è stata la Video Brain, una azienda costituita quasi tre anni fa per la progettazione e lo sviluppo di personal computer. L'azienda aveva raggiunto una buona notorietà per aver sviluppato una tecnologia del colore particolarmente invidiata dalla concorrenza. Tanto che la Radio Schack, una grossa organizzazione commerciale e all'avanguardia nelle vendite di calcolatori personali (TRS-80), aveva contattato e firmato con la Video Brain un contratto perchè questa, utilizzando la sua tecnologia dei chip, progettasse un apparecchio low-end vendibile a meno di 500 dollari. Il valore del contratto superava i 5 milioni di $ e prevedeva già per quest'anno una fornitura di 15 mila sistemi. Invece che rilanciare l'azienda verso nuove mete, quest'accordo rappresentò per la Video Brain il punto di rottura. Da un migliaio di persone l'organico in breve tempo discese a poche decine di unità obbligando il poco avveduto management ad abbandonare la fabbrica di Santa Clara (in affitto da un anno) per traslocare in un modestissimo impianto di Palo Alto. Non bastasse questo, la Video Brain è stata pure citata in giudizio da due aziende di Hong Kong per inadempienze contrattuali con la richiesta di un risarcimento danni per 10,5 milioni.

Minitraduttori elettronici a ruba

Nelle scorse settimane è iniziata in Francia la commercializzazione di un piccolo apparecchio tascabile per la traduzione elettronica di un gruppo limitato di parole (circa un migliaio). L'unità si chiama FA 300 ed a realizzarla è stata la Friends/Amis, una giovane azienda con una sede a Parigi e l'altra a Redwood City (California). La Friends/Amis è una società costituita da due francesi - Jean François e Jacky Gaillard - e un americano - M. Ron Gordon. Il prodotto ideato, sviluppato e costruito dalla Friends/Amis (della vendita si occupa la Craig) rappresenta l'ultimo nato della rivoluzione microelettronica. Il sistema si chiama FA 300, traduce parole e frasi confezionate da una lingua all'altra (la stessa denominazione sottintende l'oggetto sociale: Friends-Amis). Come luogo dove svolgere l'attività è stata scelta la Silicon Valley; il minitraduttore si compone di un microprocessore della Mostek, di un display importato dal Giappone, e di un contenitore acquistato da un fornitore di Hong Kong. Complessivamente la Friends-Amis conta di vendere quest'anno 400.000 apparecchi. L'azienda, per svilupparsi, ha bisogno di mezzi finanziari da destinare in particolare alla messa a punto di minimoduli intercambiabili. Nell'immediato una parte di queste risorse la Friends/Amis se le procurerà cedendo la licenza di costruzione. Un accordo è già stato definito con un gruppo giapponese.

Aggiungi...
un pezzo al tavolo

Siamo in una nuova dimensione nei terminali conversazionali: con la maggior parte delle caratteristiche che di solito si trovano nei terminali video di prezzo più elevato. Dotati di velocità e affidabilità gli Hazeltine hanno capacità specifiche per compiti specifici, con tutte le caratteristiche richieste per il «data enquiry» ed il «data entry». Studiate in ogni particolare per offrire la massima efficienza ed il massimo confort all'operatore, sono inoltre predisposti con la tastiera numerica separata (tranne il modello 1400) per rendere le entrate numeriche più veloci, più facili e senza errori. Lo schermo di 12 pollici, progettato e costruito dalla Hazeltine, come ogni parte di video, è il risultato di mezzo secolo di esperienza che ha permesso di unire ad una eccellente geometria, un basso consumo. Il terminale, i cui componenti basati su microprocessore sono integrati su un unica scheda, è stato studiato per ottimizzare le operazioni interattive in tempo reale. È inoltre possibile il collegamento locale e remoto per mezzo delle interfacce EIA RS 232 e CL 20mA. I modelli con cui la Hazeltine si presenta sono: 1400, 1410, 1500, 1510, 1520, dove i numeri indicano prestazioni crescenti: per il 1520 ad esempio esistono 128 tasti funzione, format-mode, comandi per l'editing, interfaccia in uscita parallela e seriale. Hazeltine: la scelta migliore al minor prezzo.

...e in più vi diamo una mano grossa così.

Hazeltine
è rappresentata in Italia da:

20124 MILANO - Via Timavo, 12
Tel. (02) 6073184 60773255 6070088

Desidero ricevere informazioni su Hazeltine
Ditta ....
Nome ....
Via ....
Tel. ....
Cap ....
Città ....

La NEC produrrà μP in Europa

Si fanno sempre più insistenti le voci, mai smentite dall'interessata, di un secondo insediamento industriale della NEC Electronics in Europa in aggiunta all'ampliamento dello stabilimento irlandese nel quale attualmente lavorano circa 150 persone. La NEC Electronics è una consociata al 100% della giapponese Nippon Electric (70.000 persone e 700 miliardi di yen di fatturato nell'esercizio al 31 marzo scorso), molto attiva in Europa nel settore dei componenti elettronici. Recentemente è stata ufficializzata la presenza anche nel nostro Paese attraverso la costituzione di una propria affiliata. Fino ad oggi la penetrazione nel Vecchio Continente è vissuta per lo più su componenti passivi e memorie per utenza EDP. Il grosso dell'attività viene svolta con memorie RAM dinamiche da 16 K per le quali la NEC è seconda fonte della 4116 progettata dalla Mostek. Ora il gruppo nipponico vuole «sfondare» anche nel segmento dei microprocessori, mettendo a frutto alcuni sviluppi particolarmente originali che già gli hanno permesso di realizzare dei μP 8 bits per applicazioni consumer e μP a 16 bit come il μCom 1600. Come le memorie anche i microprocessori verranno prodotti in Europa.

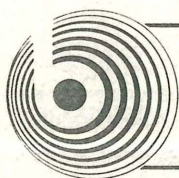
Verso la specializzazione e la segmentazione

Con l'annuncio da parte della Texas Instruments del suo primo personal computer (il TI 99/4, 1150 dollari di costo, compresi monitor e tastiera) il mercato di questi sistemi acquista ancor più vitalità e credibilità. Nello stesso tempo il mercato da atto della convergenza dei costruttori verso la specializzazione e verso la segmentazione. Una prima ripartizione riguarda il personal computer e l'home computer. Anche l'area delle applicazioni gestionali sta interessando i fabbricanti di sistemi a microprocessore; alcuni, come la Apple Computer, stanno assumendo un orientamento abbastanza marcato in tal senso. Ecco quindi perchè quanto prima società come IBM, Hewlett-Packard, DEC e Data General dovrebbero abbandonare la riserva ed entrare in campo.

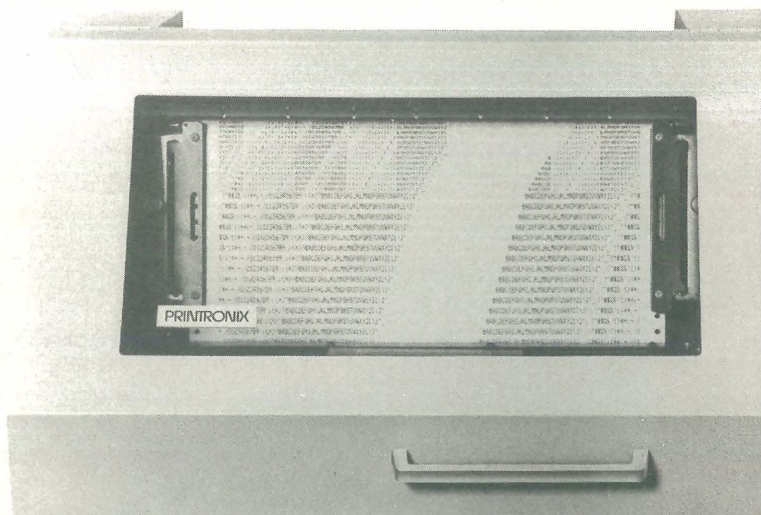
La Siemens spende per la R&S l'8% del fatturato

Nella ricerca e sviluppo la Siemens spende circa l'8% del fatturato, una percentuale che la allinea con la media mondiale delle società tecnologicamente all'avanguardia. Nell'ultimo esercizio ciò ha significato investimenti per 2,3 miliardi di DM. Complessivamente il gruppo impiega nella R&S circa 27 mila persone. Ciò ha permesso di aggiornare e ringiovanire il parco prodotti: il 45% del fatturato viene fatto con prodotti con meno di 5 anni di vita ed il 30% con prodotti introdotti da 50 a 10 anni fa. Fra i settori che hanno largamente beneficiato di questo sforzo figura l'informatica, la cui divisione dà lavoro a 16.800 persone e per la quale sono stati spesi nel 78 per R&S circa 400 milioni di DM. Il software ha avuto nettamente la precedenza, assorbendo il 60% dell'intero budget divisionale di R&S.

Un home computer belga

Al Compex 79 di Bruxelles una società belga ha presentato un home computer di sua progettazione. Il costruttore in questione si chiama Data Application International. L'apparecchio ne prende le iniziali. Il DAI si basa su un microprocessore 8080A (più ROM fino a 22 kbytes e RAM per una capacità massima di 48 kbytes oltre a circuiti I/0) e si differenzia da altri apparecchi del genere perchè visualizza sia in bianco/nero che a colore. Il costo sfiora i 1000 dollari circa. Tale è il prezzo richiesto dalla Teleac, una organizzazione di corsi televisivi sulla programmazione dei microcomputers, principale acquirente di DAI (che rivende naturalmente ai frequentatori dei suoi corsi).

Aggiungi... un pezzo al tavolo

Perchè la stampante lineare a matrice? Perchè è affidabile, veloce, silenziosa. Printronix è anche di più. I sets intercambiabili permettono alla Printronix una versatilità unica: block-mode, bar-code, caratteri a doppia altezza, sono solo alcune delle possibilità di questa macchina che può lavorare anche in Plot-mode senza alcuna modifica Hardware. Una tecnica molto avanzata, ma semplicissima, ha ridotto le parti meccaniche in movimento del 50% rispetto alle tradizionali stampanti, permettono inoltre una perfetta scrittura. La decodifica dei caratteri affidata a Proms residenti sono disponibili fino a 160 simboli: 96 ASCII più 64 a vostra scelta.

Quest'ultima è praticamente illimitata perchè la Printronix utilizza il «block-mode» stampa punti in qualsiasi posizione del foglio. L'elevata affidabilità della stampante riduce al minimo il tempo medio di riparazione (MTTR = 30 minuti). La gamma Printronix comprende i tre modelli P 150, P 300, P 600, linee al minuto perfettamente identici tra di loro. Printronix: 150, 300, 600 linee al minuto di caratteri nitidi e chiari.

...Printronix: e in più vi diamo una mano grossa così.

PRINTRONIX

è rappresentata in Italia da:

20124 MILANO - Via Timavo, 12
Tel. (02) 6073184 60773255 6070088 692882

Desidero ricevere informazioni su Printronix
Ditta ....
Nome ....
Via ....
Tel. ....
Cap ....
Città ....

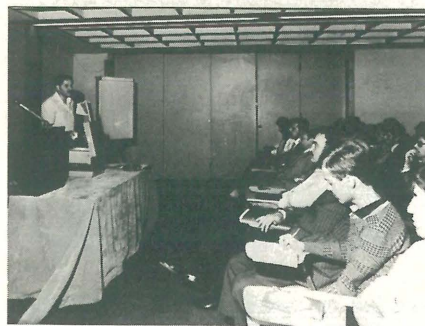
BIT '79:
una mostra già troppo stretta

a cura della Redazione

Sono stati quattro giorni entusiasmanti. La prima rassegna del personal computer, organizzata dal nostro gruppo editoriale e dall'International Marketing Center, ha avuto un successo che è andato al di là di ogni previsione, confermando la vastità di un fenomeno e l'interesse per un settore che, benché nuovo, è ricco ormai di numerosissimi prodotti (per la maggior parte di importazione, evidenziando anche in questo caso una caratteristica marcata del settore elettronico) e che comunque vede numerose aziende italiane presenti non solo nell'attività di commercializzazione, ma anche in quella di realizzazione.
La rassegna è quindi stata la prima verifica della validità di un settore in cui BIT come iniziativa editoriale è stata la prima a credere e che, nei limiti delle sue possibilità, ha contribuito a diffondere. Ed è stata un successo anche per le aziende che hanno partecipato, le quali hanno confermato una presenza di visitatori al di sopra delle loro previsioni e che nella pratica ha voluto dire il restare a secco di dépliants negli ultimi giorni e l'essere assediati in stands rivelatisi di conseguenza troppo piccoli.
A battenti chiusi, occorre fare delle considerazioni, la prima delle quali è quella di evidenziare che chi vedeva nel personal computer il giocattolo da regalare a Natale, ebbene è stato smentito.
La convinzione degli operatori del settore e nostra, confermata, qualora ce ne fosse bisogno, da quanto si è visto ed ascoltato nei giorni della rassegna, è l'interesse o meglio l'urgenza, da parte di un'utenza molto vasta e di varia estrazione, verso strumenti di elaborazione di costo contenuto, di immediata applicazione, di semplice operatività. Il rapportare queste istanze ai prodotti, però, non è altrettanto immediato, soprattutto quando si considera che un'utenza di massa è portatrice di richieste estremamente diversificate se non addirittura incerte, in ciò stimolata da un errato concetto di strumenti di elaborazione a basso costo, o più generalmente di computers.
Oggi comunque la presenza di prodotti di differenti prestazioni e le iniziative ormai capillari di diffusione degli stessi, senz'altro contribuiscono ad individuare con maggior cognizione i settori di applicazione; d'altra parte l'interesse ufficioso verso questa fascia d'utenza da parte delle grosse società EDP (IBM, Hewlett-Packard, etc.) è indicativo delle prospettive verso cui ci si muove.
Il recente annuncio della Texas Instruments, che ha finalmente presentato in America il suo home computer, è, pensiamo, il primo passo.

BIT 79 è stata il battesimo ufficiale del personal computer in Italia. Il pubblico è stato quello delle grandi mostre: è venuto da tutta l'Italia. I visitatori registrati all'entrata sono stati 3000, ma i vari espositori hanno distribuito dépliants per un numero notevolmente superiore: di fatto si era un po' stretti, e si vede.

Nel corso della rassegna è stata tenuta una serie di conferenze con l'intento di chiarire le varie applicazioni dei sistemi a microprocessore, con particolare riferimento al personal computer. Il convegno è durato quattro giorni, come la rassegna, ed è stato articolato nelle seguenti sezioni: il microelaboratore 1) alcuni aspetti metodologici; 2) applicazioni professionali; 3) la didattica; 4) oggetto di largo consumo. Nei prossimi numeri pubblicheremo gli interventi dei relatori. Nelle foto alcuni momenti del convegno.

BIT 79 non ha avuto il beneficio di un annuncio equivalente (d'altra parte l'occasione in cui è stato presentato l'home computer TI è stato il recentissimo Consumer Electronics Show, tenutosi a Chicago negli stessi giorni di BIT 79); ciononostante i prodotti presentati sono stati una novità per la maggior parte dei visitatori, che hanno avuto fra l'altro l'occasione di verificare praticamente le capacità dei vari sistemi.
Si è detto all'inizio che BIT 79 è stata la prima verifica di una nuova utenza dei sistemi a microprocessore e che in tal senso è stata positiva. È chiaro comunque che l'ampiezza di questo settore è già tale che BIT 79 non ha potuto accogliere tutte le aziende che vi intervengono, soprattutto per ragioni organizzative, per cui se da una parte si è ritenuto utile avere direttamente dalla maggior parte degli operatori presenti alla rassegna indicazioni sulle proprie modalità di intervento, d'altra parte riteniamo altrettanto indispensabile allargare in seguito il discorso allo scopo di dare una immagine il più possibile ampia e completa del settore.

EDICONSULT: sistemi a pronta consegna

BIT: *La EDICONSULT è una società commerciale che presenta fondamentalmente due classi di prodotti. Quali?*

R: La EDICONSULT tratta due tipi fondamentali di microcomputers. Il primo tipo è quello del personal computer, ossia sistemi orientati verso l'utenza personale, vista come utenza domestica, hobbistica e di piccoli enti, quali lo studio tecnico, la piccolissima azienda, il negozio, e così via. Per questo tipo di utenza noi proponiamo una serie di computers sia della Ohio Scientific che di altre case.
L'altro tipo di prodotti è quello che in inglese è chiamato small business computer. È da chiarire che il mercato dei microcomputers orientati verso la piccola o la media utenza gestionale va affrontato con tecniche di commercializzazione nettamente differenti da quelle da adottare per il mercato del personal computer; ossia, a macchine diverse si associano canali di marketing diversi. Per quel che riguarda dunque il mercato dello small business o piccola-media utenza, la EDICONSULT offre tutta una gamma di prodotti, che sono validissimi ai fini della soluzione di un problema che direi ormai decennale, quello di poter offrire ad un piccolo utente un computer gestionale ad un costo compatibile con le possibilità economiche dell'azienda. In questo caso ovviamente si entra in concorrenza con quei colossi del campo dell'EDP che toccano questa fascia di mercato. La EDICONSULT, rispetto a questi colossi, offre un computer altrettanto valido, e tale comunque da presentare caratteristiche di affidabilità di software di base e di software applicativo paragonabili. Gli elaboratori che svolgono questo tipo di funzione sono chiaramente elaboratori della fascia alta dell'Ohio Scientific (cioé la serie Challenger 3), oppure elaboratori Alpha Microsystems, Cromenco e aziende similari.

BIT: *Qual è il supporto fornito?*

R: Per quanto concerne il personal computer, ci presentiamo all'utente finale tramite una rete dettagliata di rivenditori in tutta Italia. Questa rete di rivenditori è in grado di fornire assistenza tecnica e assistenza software. In tal modo riteniamo di poter fornire insieme al sistema un pacchetto completo di assistenza. Pertanto i nostri clienti hanno la garanzia di acquistare un computer che ha non solo una validità attuale, ma anche una validità futura.
Nello stesso tempo siamo presenti a livello internazionale nel campo dell'acquisizione e in quello delle ricerche di mercato, onde poter

fornire sempre dell'hardware e del software adeguati ai progressi del momento in cui vendiamo.
Per quel che riguarda invece lo small business computer, che poi in effetti non è destinato solo alle piccole imprese, ma anche alle medie, noi offriamo i computers di quattro grandi case (se andassimo oltre, non potremmo fornire un'assistenza adeguata) e teniamo un contatto diretto con le case costruttrici per poter continuamente aggiornare sia l'hardware che il software.
Anche per gli small business computers raggiungiamo l'utente finale tramite concessionari, che in questo caso sono spesso delle software houses, piuttosto che dei semplici rivenditori. Queste case di software contribuiscono alla personalizzazione di pacchetti di software applicativo orientandolo specificamente verso le esigenze dell'utente finale.
In questo campo annunciamo una politica tipica della EDICONSULT: noi non venderemo esclusivamente macchine, ma venderemo software di base e software applicativo insieme alle macchine. I nostri concessionari avranno un insieme completo di hardware, di software di base e di software applicativo da offrire, garantendo così all'utente finale un lavoro coordinato e standardizzato, sia in campo fiscale che nel campo delle proceduristiche aziendali.

BIT: *Qual è la sua opinione su questo fenomeno?*

R: Attualmente, tanto in America, dove è sorto il fenomeno del microcomputer, che in Europa, e in particolare in Italia, il mercato è molto confuso, in quanto accanto ad aziende molto serie operano aziende che si pongono in un modo che secondo me non è tale da risolvere il problema del cliente. Addirittura, e non è un 'affermazione mia, esiste una vera e propria pirateria nel settore del microcomputer. Quindi tengo a ribadire che la EDICONSULT non intende fare della vendita pirata, ma una vendita onesta, una vendita corretta. È per questo che ci preoccupiamo di assicurare al cliente, primo, una distribuzione di prodotti validi, secondo, la certezza che i prodotti venduti saranno garantiti con una costante assistenza, una distribuzione continua di parti di ricambio e una continuità di fornitura del software, per cui potranno essere funzionanti in ogni momento anche nel futuro. Il grosso problema oggi nel settore del microcomputer è che su 20 o 30 aziende che costruiscono microcomputers sono da considerare valide solo 4 o 5, cioè quelle che non solo forniscono un computer completamente valido sul piano hardware e sul piano del software di base e applicativo, ma nello stesso tempo forniscono pezzi di ricambio affidabili e nella quantità giusta per poter assistere un mercato straniero.
La EDICONSULT pertanto si impegna ad un servizio completo di assistenza, garantendo un hardware che può essere rifornito di parti di ricambio anche fra due, tre, quattro o cinque anni, anche dopo che l'hardware stesso sia stato modificato.
Un'ultima cosa, come amministratore della EDICONSULT tengo a ribadire: noi riteniamo che un'azienda, per poter lavorare in questo settore in modo da riuscire a svolgere una funzione valida per l'utente finale, debba avere, sia dal punto di vista finanziario che da quello degli impianti, dimensioni che le permettano di tenere a magazzeno tutta la fascia di computers che si devono vendere, perché solo così è possibile garantire un'assistenza completa e una distribuzione efficace. Oggi, la EDICONSULT si vanta di essere probabilmente l'unica azienda in questo campo che può mostrare un magazzeno pieno non solo di parti di ricambio, ma di computers pronti per la consegna.

Intervista al Dott. Lodà - Amministratore Unico della Ediconsult.

HARDEN: un'organizzazione efficiente

BIT: *La HARDEN è stata la prima azienda italiana che ha introdotto il personal computer in Italia, e più precisamente il PET della Commodore. E' naturale, pertanto, chiedere alla HARDEN di fare il punto su questo fenomeno.*

R: Quella che, all'inizio del 1978, sembrava un'avventura, e a dispetto di chi considerava il personal computer poco più che un giocattolo, sta rivelandosi invece un'operazione utile per tutto questo mercato. Prima di tutto perché il personal computer in generale, e il PET in particolare, risulta tutt'altro che un giocattolo, e in secondo luogo perché il personal computer ha oggi raggiunto un livello tale che consente finalmente di dotare piccole aziende, professionisti ed altri tipi di utilizzatori di un mezzo valido per risolvere i loro problemi di lavoro, ad un prezzo accessibile.
Il PET della Commodore è indubbiamente un computer che non solo ha fatto parlare di sé, ma che offre un rapporto prezzo/prestazioni estremamente interessante. La Commodore, che fino al 1976 aveva prodotto solo beni di consumo nel settore elettronico, per prima nel mondo ha avuto l'idea di costruire qualche cosa di più di un semplice calcolatore portatile più o meno sofisticato, vale a dire di creare un computer da vendere a livello consumer. Il successo è sta-

to strepitoso, ed ora la Commodore sta potenziando la sua produzione per soddisfare pienamente le esigenze dei paesi europei: chiaramente in questo ha dovuto dare la precedenza ai due mercati maggiori, cioè a quello inglese e a quello tedesco. Però abbiamo anche l'orgoglio di affermare che dopo quello inglese e quello tedesco viene il mercato italiano. Infatti noi in questa mostra esponiamo la nuova serie 3001 dotata di floppy-disk e stampanti Commodore, avuta immediatamente dopo Inghilterra e GQRMA-dopo Inghilterra e Germania, mentre invece altri paesi europei dovranno attendere settembre. uesto ci fa molto piacere, perché come HARDEN siamo riusciti a dimostrare alla Commodore una sufficiente intraprendenza commerciale, che ci ha consentito di metterci al terzo posto in Europa.
Per concludere, ritengo che molti problemi a livello professionale, a livello industriale, a livello commerciale possano essere risolti con il personal computer, in quanto esso fornisce ad aziende, ad operatori tecnici, a ingegneri, architetti, geometri, avvocati, medici, etc., uno strumento di facile ed immediata utilizzazione che smitizza tecnicamente e come costo il mondo dell'informatica.

BIT: *Qual è la vostra strategia nella distribuzione del PET?*

R: Devo premettere innanzitutto che la Commodore dà ai suoi distributori nazionali delle precise disposizioni alle quali gli stessi distributori, e noi in particolare per l'Italia, devono attenersi. Il traguardo della Commodore è quello di avere entro il 1980 almeno un distributore in ogni capoluogo di provincia. Attualmente la rete costituita dalla HARDEN tocca tutte le regioni; in particolare i distributori della HARDEN sono 47. Non abbiamo ancora tuttavia assolto all'impegno preso con la Commodore di avere un distributore in ogni capoluogo di provincia. Diciamo che la distanza media che separa oggi un nostro centro di vendita dall'utilizzatore non supera i 100 Km. Chiaramente, quando avremo completato la rete di distribuzione, il fatto di essere presenti in ogni capoluogo di provincia risolverà anche i problemi di capillarità in ordine sia alla distribuzione sia all'assistenza software e hardware.
Infatti ogni distributore deve avere non solo una struttura caratterizzata da una evidente proiezione di carattere commerciale, ma anche la capacità di fornire supporto di software e di hardware. Quindi il nostro principale sforzo è attualmente quello di consolidare e migliorare i requisiti che ogni distributore deve avere.
Perché questa impostazione capillare, e soprattutto organizzata in questo modo? In primo luogo perché prevediamo una diffusione di massa, in secondo luogo perché i distributori locali siano nelle condizioni di soddisfare meglio e più puntualmente la possibile clientela.
Pertanto provvediamo a preparare tecnicamente il personale dei nostri distributori perché sia in grado di rispondere prontamente e con competenza alle richieste di assistenza tecnica o di programmazione. Inoltre i programmi sviluppati dai vari distributori locali sono centralizzati presso un nostro ufficio, il quale provvede a distribuirne la lista a tutti gli altri, per cui se, ad esempio, a Reggio Calabria è stato già realizzato un dato programma, si evita di sprecare risorse sullo stesso problema.
Questo permette ai nostri distributori di intervenire tempestivamente in tutti quei campi che prima ho citato.
Riteniamo che una organizzazione, impostata nel modo che ho ricordato e capace di assolvere tempestivamente alle problematiche più varie, rappresenterà molto presto un esempio più unico che raro anche in Italia, in quanto non mi risulta che esista un'organizzazione che possa coprire un fronte così vasto di necessità, e che sia capillarmente diffusa anche a livello di assistenza software e hardware.

BIT: *Quali sono oggi i settori di impiego del PET?*

R: Con l'avvento della nuova serie 3001, e soprattutto con l'avvento dei floppy-disk e delle stampanti, le problematiche che prima richiedevano delle soluzioni "acrobatiche", oggi sono risolte con minore impegno, con più elasticità, con maggiori possibilità.
Il PET permette inoltre ulteriori prestazioni dovute al fatto che ha un bus di uscita totalmente gestito dal computer stesso, sia come input, sia come output; in aggiunta ha anche un real time clock anch'esso accessibile via software. Delle numerose applicazioni fatte sfruttando queste caratteristiche, mi limiterò ad alcune solamente. Ad esempio il PET è stato utilizzato per gestire strumentazione estremamente sofisticata. Ci sono quindi applicazioni a livello medicale, di controllo di reti elettriche tramite una serie di strumenti (voltmetri, frequenzimetri) continuamente monitorati e con registrazione periodica dei dati oppure al verificarsi di condizioni anormali. Un'altra applicazione di cui abbiamo notizia è l'utilizzazione del PET per gestire lavori di macchine automatiche.
Un'ulteriore applicazione si è avuta nel settore dei radioamatori, vale a dire il PET si sintonizza automaticamente ad un determinato orario (attraverso il real time clock) con il radioamatore di una città lontana, registrando il messaggio ed eventualmente lasciandone un altro precedentemente registrato.
Queste applicazioni danno la misura della versatilità di questo computer. Chiaramente non pensiamo che queste siano le applicazioni principali. Certo è che anche in questi campi finalmente anche l'elettronica ha messo a

disposizione un mezzo fino ad ieri sicuramente impensato ed impensabile.

Intervista al Sig. Bonezzi - Direttore Commerciale della Harden - Divisione elettronica.

HOMIC: sfatare un mito

BIT: *Con quale filosofia la HOMIC, come società commerciale, affronta il discorso del personal computer?*

R: Il nostro intento è quello di diffondere, per quanto possibile, l'uso e l'impiego di questi microcalcolatori, di sistemi, cioè, estremamente capaci, molto versatili, di prezzo contenuto e che possono quindi raggiungere un pubblico vastissimo che al momento non sa ancora che cosa può fare con queste macchine e che è convinto che siano macchine inavvicinabili sia dal punto di vista operativo che dal punto di vista prezzo. Il nostro scopo è cercare di sfatare questo mito, fornendo macchine di prezzo contenuto, molto semplici da usare, che quindi possono essere usate direttamente dall'acquirente.
Questo nostro approccio è evidenziato dai prodotti che distribuiamo in esclusiva e che sono l'SwTPC (che è una macchina di un produttore americano per applicazioni tipicamente di carattere gestionale), il TRS-80 della Radio Schack, ed il Nascom, che è un piccolo calcolatore fornito o in kit o già montato.
La nostra politica commerciale è in relazione alle caratteristiche ed alla espansione che il mercato del personal computer avrà ed è perciò quella di non creare una rete di distribuzione esclusiva, ma di avere fornitori a livello OEM oppure di rappresentanza che perseguano dei canali ben precisi, cioè persone che abbiano una competenza tecnica per settori di applicazione e di conseguenza la possibilità di intervenire a livello capillare in quel settore con questa ottica stiamo creando una rete di distribuzione che copre praticamente tutto il territorio. Comunque oggi abbiamo distributori per la Sicilia e la Calabria, un distributore a Roma, uno a Udine, uno a Modena, uno a Genova e uno a Torino. Altri distributori coprono, insieme a noi, la zona Milano, nord Milano, che è la zona più estesa con un mercato vastissimo, e quindi tutto l'hinterland milanese.

BIT: *Chi possono essere gli utenti di questi sistemi?*

R: L'esperienza di questi mesi ci porta a dire che gli utenti possono essere tutti, senza nessuna distinzione. Innanzitutto c'è il "livello casa", ossia una persona che vuole una macchina per tenere la contabilità domestica, l'elenco telefonico, il registro delle scadenze, o che al limite vuole giocare. A questo punto noi vogliamo portare avanti, e speriamo moltissimo di aver successo, il discorso dell'apprendimento, cioè la macchina che aiuta il ragazzo a creare veramente, ossia non solamente a giocare, ma a creare dei programmi di apprendimento e di evoluzione. Altri livelli di applicazione sono tutti i campi del commercio, delle attività tecniche, delle attività scientifiche. Come esempi, anche se particolari, sono da citare inoltre le applicazioni di questi sistemi in laboratorio in cui il sistema è collegato a strumenti di misura con elaborazione dei relativi dati, oppure per la gestione di canali telefonici per radio private, o ancora come titolatrice per TV private. È difficile comunque dire ora esattamente chi è l'utente tipico. Dalle domande che ci vengono poste, diciamo tutti.

BIT: *Come la HOMIC affronta il problema dell'assistenza?*

R: È un problema certamente molto sentito, e direi che la maggior parte delle domande del cliente riguardano il problema dell'assistenza. Le nostre macchine sono innanzitutto fornite con una garanzia di tre mesi che è una garanzia totale della macchina. Forniamo inoltre dei contratti di assistenza e di manutenzione di durata annuale che garantiscono all'utente finale di avere una macchina sempre funzionante. Possono essere dei contratti di tipo preventivo oppure dei contratti su richiesta. Abbiamo poi un altro tipo di contratto, che noi chiamiamo di pronta sostituzione: data la modularità delle macchine, è fatta la sostituzione del pezzo avariato, che può essere la scheda, o il singolo componente, per poi ripararlo in laboratorio.

Intervista al Sig. Franchini-Direttore vendita della Homic.

PLAE: un nome nuovo

BIT: *La PLAE è un nome non molto noto nel campo del personal computer, per cui è d'obbligo saperne di più e come azienda e come prodotti.*

R: La PLAE è una giovane azienda nata alla fine del 1977 come settore staccato della ELPA, nome che crediamo sia più conosciuto, in quanto come ELPA operiamo da oltre 20 anni nel campo dell'elettronica industriale (automazione, telecomunicazioni, etc.) in Italia e all'estero.
La nascita della PLAE risponde alla logica di affrontare in maniera specializzata il settore dei microprocessori e delle relative applicazioni industriali, avvalendoci in questo del notevole background acquisito come ELPA. È chiaro quindi che l'intervento nel campo delle realizzazioni di microcomputers è stato un obbiettivo abba-

stanza naturale da perseguire, e di fatto oggi siamo in grado di presentare un nostro personal computer, chiamato DELTA 1, per venire incontro alle esigenze di questo settore con un prodotto valido e competitivo.
Il DELTA 1 è stato costruito attorno alle schede della Synertek (quindi ha come CPU il 6502), sulle quali siamo intervenuti per fornire prestazioni che sono nettamente superiori a quelle delle singole schede. Inoltre abbiamo curato con particolare attenzione l'aspetto estetico, e di fatto il sistema presenta una veste che permette un inserimento gradevole sia in un ambiente professionale che in un ambito domestico: in particolare può essere inserito in una valigetta ventiquattrore, dato il profilo molto piatto, come una modernissima macchina da scrivere.
L'interfaccia verso l'utente è costituita da una tastiera ASCII di 54 tasti per 126 caratteri (minuscole, maiuscole, segni d'interpunzione e speciali) e da un TV domestico collegabile direttamente in antenna. Per usi più professionali è possibile l'interfacciamento ad un monitor TV. È fornito quindi di un connettore DB 25 per l'interfacciamento con terminali EIA RS 232 (ad es. una stampante seriale), cioè i più universali possibile.
Infine come periferica di massa abbiamo ovviamente utilizzato la cassetta magnetica per ragioni di economicità (è controproducente inserire periferiche di costo superiore a quello della CPU) e di mercato, in quanto il DELTA 1 è un personal/home computer destinato ad un largo pubblico.
È chiaro comunque che questo sistema rappresenta un punto di partenza per ulteriori sviluppi, per cui la PLAE si sta dedicando, oltre che alla produzione del DELTA 1, che già assorbe una notevole parte delle sue capacità produttive, anche allo sviluppo di sistemi gestionali per operatori commerciali, punti di vendita, etc.

La politica della PLAE è quella di seguire con estrema attenzione e con estrema professionalità le richieste del mercato.

Intervista al Sig. Paderni - Amministratore Unico della PLAE.

UNICOMP: i personal professionali

BIT: *La UNICOMP ha del mercato Personal Computer un'idea estremamente precisa ma altrettanto differente da quella delle altre società operanti in questo settore. Che cosa propone e a quale utenza?*

R: La UNICOMP si presenta essenzialmente con dei prodotti orientati più che al mercato hobbistico, al mercato della piccola azienda, del commercialista, dell'avvocato, del grosso negozio, cioè di chi vuole impiegare il personal computer o come strumento di gestione, o come strumento di calcolo, ma in ogni caso principalmente come strumento professionale.
Noi riteniamo che il grosso del mercato risieda nelle applicazioni di tipo gestionale, nelle applicazioni di tipo tecnico (calcolo tecnico-scientifico, di struttura, calcoli termici, etc.).
È possibile che il mercato dell'hobbistica o dell'impiego personale si sviluppi e acquisti delle dimensioni veramente ragguardevoli; noi però riteniamo che in questo momento la change maggiore stia nel mercato professionale e nel mercato dell'applicazione gestionale.
I prodotti che la UNICOMP ha derivano da un'esperienza di quasi un anno di lavoro. Sono prodotti passati attraverso un vaglio di prove pratiche, di valutazioni tecniche e di valutazioni teoriche (sia di affidabilità, sia di qualità software). L'idea originale di restare sul mercS-100 e di compatibilità S-100 rimane ancora uno dei punti più validi. Questo è dovuto alla compatibilità software, cioè alla disponibilità di un sistema operativo, il CP/M, che ha un eccellente corredo di linguaggi, (COBOL, FORTRAN, BASIC), il che permette di intercambiare in modo immediato i programmi su tutte le macchine di questa linea.
I prodotti che attualmente la UNICOMP maggiormente spinge e commercializza sono il Sorcerer della Exidy che ha il vantaggio di nascere anche come sistema hobbistico, ma che però può crescere anche come sistema hobbistico, ma che però può crescere e diventare un piccolo sistema gestionale, professionale; l'Horizon della North Star, che è un sistema che noi riteniamo di eccellente qualità sia per le caratteristiche intrinseche della macchina (velocità di elaborazione, capacità di archiviazione, affidabilità della macchina), sia per il corredo software (proprio in questi giorni è stato annunciato dalla North Star il linguaggio Pascal). A livello maggiore si trova il Cromenco, che ormai è arrivato alla soglia del minicomputer. Sul Cromenco è stato

annunciato il disco da 10 Megabyte, quindi può montare sia floppy IBM compatibili, sia unità a dischi in tecnologia Winchester con dei costi veramente competitivi. Questo per quanto riguarda i prodotti.
Per quanto riguarda l'approccio commerciale, noi riteniamo che il miglior canale di commercializzazione di queste macchine sia il negozio: ossia una organizzazione che abbia un suo mercato locale e che quindi sia conosciuta localmente, permettendo così di mantenere bassi i costi di commercializzazione.
Questo è un primo punto. Un secondo punto riguarda l'assistenza. Per quanto questi sistemi siano di costo contenuto, essi vanno in un ambiente in cui diventano uno strumento di gestione e di operazioni vitali, e quindi richiedono un'assistenza praticamente dello stesso livello dei grossi computers. L'azienda che faccia su una macchina di questo genere una contabilità generale, una fatturazione, etc., deve avere la macchina funzionante, deve avere dei programmi ragionevolmente buoni e deve avere tutte quelle caratteristiche che si trovano normalmente su computers più grandi. Pertanto noi riteniamo fondamentale per un discorso personal computer applicato alla gestione una asssistenza capillare, brevi tempi di intervento, buone risorse software.

BIT: *La UNICOMP fornisce, insieme al sistema, software applicativo?*

R: Nell'ambito dell'intervento nell'ambiente gestionale, la UNICOMP ha ritenuto di dover sviluppare, utilizzando il know how aziendale, pacchetti software gestionali. Noi riteniamo che il costo dello sviluppo di un buon pacchetto software - e con buon pacchetto software intendo essenzialmente un pacchetto che sia affidabile, facilmente personalizzabile, facilmente modificabile, che consenta veramente di risolvere un problema aziendale

- sia decisamente troppo alto per un piccolo negozio o per un piccolo dealer.
Un buon pacchetto software richiede un know how aziendale consolidato e richiede un'esperienza, delle risorse di programmazione e dei costi che vanno al di là di quello che può essere gestito profittevolmente su un piccolo numero di installazioni. Perciò noi abbiamo ritenuto di sviluppare un sistema di gestione aziendale (contabilità generale, fatturazione, gestione magazzino) che offriamo come pacchetto gestionale. Questo anche perché valutiamo che, mentre la vendita di packages è abbastanza difficile su macchine di alto valore e di elevato costo, a causa delle applicazioni che cambiano notevolmente da azienda ad azienda, su questo livello di macchina la omogeneità delle applicazioni è abbastanza alta per poter offrire, insieme al sistema, dei pacchetti di software facilmente personalizzabili. Quindi noi riteniamo che la fornitura di software applicativo sia una parte fondamentale e vitale del nostro programma di vendita nell'ambiente gestionale, perché permette ai nostri agenti di partire immediatamente, di evitare quei costi di sviluppo che secondo noi sonveramente molto alti.
Pertanto nella nostra filosofia di vendita il nostro distributore o il nostro rivenditore non è semplicemente un distributore. Non vende soltanto i prodotti, ma è in grado di fornire un'assistenza completa, vale a dire sia un'assitenza hardware, sia un'assistenza software. Un nostro distributore deve essere in grado di dare in primo luogo una applicazione chiave in mano, e deve essere in grado di fornire successivamente, man mano che le esigenze del suo cliente si ampliano, il necessario supporto.

BIT: *Dove siete presenti come rete di distribuzione?*

R: Attualmente siamo presenti in tutto il nord Italia, in parte del centro Italia ed in Sicilia. Il nostro piano è di coprire interamente l'Italia entro il 1979. Chi ha interesse ai nostri prodotti? Moltissime organizzazioni; le organizzazioni nascenti, o in modo molto più attivo i rivenditori di macchine per ufficio, i quali già si indirizzano al mercato adatto (essendo già in contatto con la clientela adatta a recepire questo nuovo tipo di prodotto) e quindi possono svolgere un'attività di mercato veramente notevole e penetrante.

Intervista all'Ing. Focardi - Direttore Commerciale della Computeria

SKYLAB : una proposta differenziata

BIT: *Qual'è la proposta SKYLAB nel settore del personal computer?*

R: La SKILAB intende operare in questo settore rivolgendosi a quei clienti potenzialmente interessati ma professionalmente poco preparati nella utilizzazione dei personal computers e dei microcomputers. Esiste una notevole curiosità nel settore, ma esiste anche un'argomentazione piuttosto generica circa i campi di applicazione. Come effettivo e tangibile esempio di applicazione, devono essere indicate non solo le applicazioni dei giochi che, per quanto interessanti, risultano a mio parere piuttosto limitative, ma anche le applicazioni gestionali, delle cui possibilità molti già si rendono conto, e soprattutto (e penso sia questa la nostra particolarità) le applicazioni di controllo industriale. A tal fine nel nostro stand presentiamo un'applicazione di controllo di un modello di

ferrovia in miniatura. Con questo esempio vogliamo dare una dimostrazione di come sia possibile effettuare un controllo industriale: i vagoni del modello possono essere considerati i carrelli di un cementificio o di una fabbrica di mangimi, le varie stazioni di lavoro di una macchina transfert. Queste sono le possibilità applicative concrete.
Attualmente noi ed i nostri consulenti stiamo lavorando ad un'applicazione molto interessante, diretta ai calzaturifici: l'ipotesi è, avendo un modello di scarpa, determinare da calcolatore tutte le misure. Questo vuol dire intervenire in un processo di lavorazione industriale in modo nuovo: nel nostro caso una volta stabilito il design della scarpa, è possibile passare alla fase esecutiva in un tempo estremamente breve. Il discorso è valido anche per tutti quei settori (ad es. quello dell'abbigliamento) in cui si lavora su una grandissima quantità di variabili che devono essere conteggiate e valutate nel più breve tempo possibile, e nei quali il contributo umano, per lavori di routine, può portare errori che si ripercuotono immediatamente sulla efficienza del ciclo produttivo.

L'attività della SKYLAB è comunque esclusivamente un'attività commerciale. Ma naturalmente ci rendiamo conto che in questo settore non è possibile proporre esclusivamente il prodotto senza dare un adeguato supporto e comunque un adeguato aiuto alla utilizzazione da parte del cliente. Pertanto la nostra attività ha sostanzialmente due canali: un'attività esclusivamente di vendita dei prodotti ad utilizzatori che conoscono già le modalità d'uso e si rendono conto delle possibilità applicative dei sistemi che noi proponiamo; un'altra proposta che facciamo è quella della realizzazione e della vendita di sistemi completi di controllo e di gestione (in ciò avvalendoci di ditte legate alla SKYLAB da particolari forme di contratto), con sviluppo del software relativo ed eventualmente di quelle interfacce e di quelle periferiche che si rendessero necessarie per la utilizzazione specifica.

Però quello che riteniamo ci possa distinguere da molte ditte che operano in questo settore è la nostra possibilità di fornire una vasta gamma di periferiche a questi sistemi. In questo settore infatti abbiamo visto un'esplosione di unità centrale a tutti i livelli, ma una grande carenza di unità periferiche che si rendono indispensabili per applicazioni di tipo industriale o di tipo gestionale e scientifico. Quindi la nostra cura è stata appunto nella ricerca di quei prodotti che potessero garantire la completa espandibilità e l'uso professionale di queste apparecchiature.

Per quanto riguarda invece l'aspetto di sviluppo software e l'aspetto di realizzazione hardware specifica, per l'impiego di tali sistemi in utilizzazioni particolari, come ho accennato prima, noi ricorriamo a ditte collegate con determinati tipi di contratti con la SKYLAB. La SKYLAB rimane però l'unica interfaccia con l'utilizzatore.

BIT: *BIT 79, prima rassegna del microprocessore e del personal computer, quali indicazioni ha fornito?*

R: L'utilità di BIT 79 è stata innegabile; la mostra è servita fra l'altro a diffondere la conoscenza di questi sistemi estremamente economici e molto potenti fra un largo pubblico. Ho notato una notevole curiosità, ma anche una sostanziale impreparazione. È indicativo il fatto che molti non si rendano conto dei problemi che possono risolvere nelle loro aziende e delle possibili utilizzazioni a livello personale. Certo una mostra di tal genere, che presentava in parallelo una serie di conferenze, è stata estremamente utile, perché ha fornito a quanti oggi sono interessati indicazioni e suggerimenti di utilizzazione di questi sistemi.

Intervista al Sig. Viero - Amministratore Unico della SKYLAB.

LA ZELCO AMPLIA LA SUA GAMMA DI PRODOTTI

La Zelco s.r.l. di Milano ha recentemente concluso un accordo di rappresentanza con la North Star Computer Inc. di Berkeley (California), per commercializzare in Italia i microcalcolatori "personal" della serie HORIZON prodotti da questa nota casa statunitense.
I calcolatori della North Star offrono le seguenti caratteristiche:
- *CPU Z80 a 4 MHz*
- *da 16 a 64 Kbyte di memoria con controllo di parità*
- *da 1 a 4 mini-floppy doppia densità (e opzionalmente doppia faccia)*
- *compatibilità totale col bus S-100 (standard universale dei personal e hobby computer)*
- *piastra matematica opzionale*
- *BASIC esteso e sistema operativo compresi nel prezzo*
- *vasto software supplementare disponibile (per es. PASCAL USDC per meno di 100.000 Lit.)*

La gamma dei prezzi OEM per un sistema provvisto di due mini floppy parte da una cifra inferiore ai 3 milioni.

Che cosa possono fare e che cosa non possono fare le macchine

di **J. Nievergelt, J.C. Farrar** University of Illinois (USA) Department of Computer Science

Introduzione

Quando ci si chiese, per la prima volta, che cosa possano fare e che cosa non possano fare le macchine, ciò non avvenne certamente in relazione ai calcolatori. Sembra che l'umanità sia stata sempre curiosa di trovare i limiti a cui può giungere la tecnologia di un'epoca. In particolare, una grande attrazione è sempre stata suscitata dall'idea di costruire macchine che possano, sotto qualche aspetto, simulare il comportamento dell'uomo. Vi è sempre stato qualche inventore che ha sognato, ha progettato, e qualche volta ha costruito macchine, quali un guerriero a vapore o uno scrivano automatico ad orologeria. Tuttavia, per effetto del grande aumento di velocità di calcolo e di complessità delle macchine, raggiunto negli ultimi decenni, e del radicale miglioramento nella nostra comprensione del modo di progettare algoritmi tali da fare sì che la macchina si comporti come vogliamo, tale domanda ha assunto importanza maggiore come non mai prima d'ora. Ed inoltre, data la sempre maggiore diffusione dei calcolatori e delle macchine comandate da essi nella nostra società è importante che il pubblico, in generale, comprenda almeno in parte le possibilità e le limitazioni di tali macchine.

La domanda: « Che cosa può fare, e che cosa non può fare una macchina? » non ammette una risposta netta, se non altro perché le parole « può fare » sono così vaghe da permettere diverse interpretazioni. Negli ultimi quarant'anni, però, alcune domande precise riferentisi ad essa sono state formulate, e alcune risposte parziali sono state ottenute. La conoscenza e la comprensione di esse può servire ad evitare confusione circa le capacità delle macchine, e a dissipare certi miti largamente diffusi.

Una macchina può pensare?

Questa è forse una delle domande più recenti e stimolanti che si possano fare riguardo a una macchina. Fino all'avvento dei calcolatori digitali, le macchine erano progettate quasi esclusivamente per effettuare operazioni puramente meccaniche, e davano perciò scarso motivo di riflessione sulle loro capacità intellettuali.

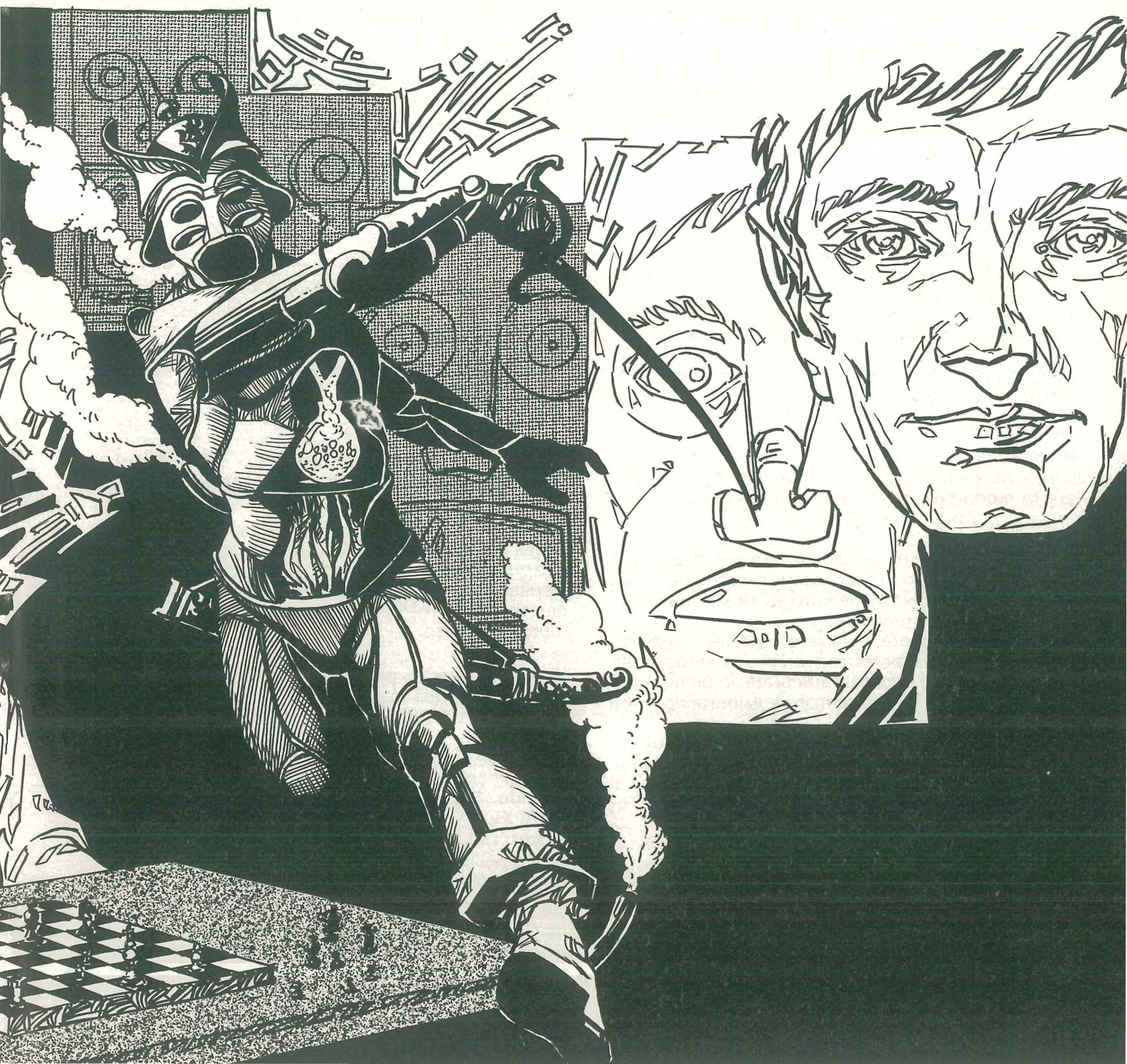
Tratto da "Quaderni d'informatica" per gentile concessione della Honeywell I.S.I.

Vi sono stati tuttavia esempi antichi che poterono far sorgere, nelle menti di molte persone, il problema della intelligenza delle macchine. Vi fu, al riguardo, il caso del falso giocatore automatico di scacchi costruito dal Barone von Kempelen intorno al 1800, che fu esposto al pubblico per molti anni, e di cui si dice che abbia vinto tutte le partite giocate. Entro la macchina era così abilmente celata una persona che, benché fosse permesso agli spettatori di ispezionare, successivamente, le diverse parti interne della macchina, egli, spostandosi dall'una all'altra parte, riuscì sempre ad evitare di essere scoperto.

Oggi esistono diversi programmi di gioco degli scacchi, che giocano al livello dei giocatori di torneo delle qualifiche inferiori, e che batterebbero la maggioranza dei giocatori occasionali. Vi è anche un programma che ha raggiunto il livello di « maestro ».

I più saranno disposti ad ammettere che questa prestazione è un risultato di natura intellettuale, non banale. Tuttavia, la discussione sulla capacità delle macchine di adempiere compiti di natura intellettuale è stata spesso oscurata da argomenti emotivi ed irrazionali.

La prova di Turing

La questione delle macchine pensanti fu discussa nel 1950 dal famoso logico inglese A.M. Turing, e, ancor oggi, la sua analisi è una delle migliori che si possano leggere da chi voglia chiarire le proprie idee al riguardo. Turing vede chiaramente che la domanda è di natura emotiva, tale da prestarsi a interpretazioni soggettive su cosa significhi « pensare », e ben presto la respinge come troppo vaga per avere una risposta significativa. Infatti, egli indica come, per certe interpretazioni della domanda: « la macchina può pensare? » la risposta non possa ottenersi con ragionamenti logici, ma solo come materia di fede: mentre, per altre, la risposta è una conclusione anticipata. (Certi argomenti circa le macchine pensanti sono basati sulla assunzione inespressa che il pensare sia, per definizione, ciò che le persone fanno: da cui si deduce che le macchine non possono pensare, a meno che si voglia prendere in considerazione la possibilità che le macchine siano persone, o che le persone siano macchine: il che non intendiamo fare).
Turing ha posto una nuova questione, al posto della

domanda originale: un modello secondo cui il «pensiero» delle macchine può essere discusso in termini puramente tecnici. Esso è noto da allora come la « prova *(test)* di Turing » ed è descritta come segue. immagini un calcolatore C, e una persona M rinchiusi in una stanza, e collegati ognuno ad una telescrivente in un'altra stanza, dove c'è un interrogatore umano, I. Il compito di I è di determinare quale telescrivente è collegata a C, e quale ad M, « battendo » domande sull'una, sull'altra, o su ambedue le telescriventi, ed analizzando le risposte che riceve. La situazione è quindi, per il calcolatore quella di una «prova » se gli si affida il compito di rendere vani gli sforzi di I: e cioé C deve rispondere alle domande di I in modo che I non possa distinguere C da M. Perciò la nuova domanda, che risulta dalla prova di Turing, e che sostituisce la primitiva: « La macchina può pensare? » è la seguente: « Vi sono macchine tali che si possa fare in modo che abbiano successo in questo gioco? ».

È chiaro che la domanda ha un carattere differente da quella primitiva, perché che cosa si intende per «aver successo » nel gioco? È una questione di grado, non materia di « sì » o « no ». Il criterio per giudicare del successo sarà probabilmente statistico: in media, l'interrogante I ha una probabilità del *p* per cento di identificare in modo corretto la macchina dopo avere interrogato per *m* minuti. Il fatto è che la prova di Turing pone problemi di progetto e di ingegneria, e in particolare di programmazione: e cioé: fino a che punto sappiamo fare una macchina che faccia questo gioco? Che ci attendiamo di saper fare tra 10 o 20 anni? Si può sempre non essere d'accordo su tali questioni, ma l'emotività di cui era carica la domanda originale è quasi completamente scomparsa.

Nel suo scritto del 1950, Turing esprimeva la convinzione che, alla fine del secolo, si sarebbero potute costruire macchine capaci di fare così bene il gioco, che un interrogatore medio non avrebbe avuto più del 70% di probabilità di giungere alla corretta identificazione dopo cinque minuti di interrogazione.

Non ci risulta che alcuno abbia tentato veramente di fare in modo che una macchina possa affrontare la prova di Turing. Si è tuttavia compiuta una buona quantità di lavoro che getta luce su come si potrebbe fare a programmare un calcolatore per la prova di Turing, e che grado di successo ci si potrebbe attendere, se un tentativo fosse fatto all'attuale stato dell'arte della programmazione euristica.

Poiché un calcolatore, per avere successo nella prova di Turing, deve poter conversare in modo sufficiente, per mezzo di parole stampate, in un linguaggio naturale, le più importanti indicazioni si ricavano da certi programmi costruiti per tenere conversazioni plausibili in ristretti campi di discorso. Discuteremo due di questi programmi scritti in anni recenti. Lasciamo che il lettore giudichi da sé, da questi esempi, quanto sia giustificata la previsione di Turing.

Programmi di conversazione

Vi sono calcolatori programmati per tenere conversazioni plausibili, e talvolta utili, in limitati dominii di discorso. Una vivace attività di ricerca continua in tale area di programmi di calcolatori, che accettano e/o rispondono in linguaggio naturale.

Lo scopo di questo paragrafo è quello di dare al lettore una certa conoscenza del modo di operare di tali programmi, in modo che egli possa farsi una idea della relazione esistente fra ciò che un tale programma può compiere e la sua complessità. Egli potrà allora congetturare se e come si possa fare, con uno sforzo concentrato, ora o fra dieci anni, un programma che possa essere sottoposto alla prova di Turing. Un programma che simuli un buon conversatore deve poter imitare due distinti modi di comportamento che si verificano nella conversazione umana. In primo luogo, esso deve capire una data affermazione o una data domanda in una qualunque delle forme in cui essa può essere formulata; deve, insomma, comprendere il significato di una frase. In secondo luogo, deve poter conversare in modo plausibile su un dato soggetto anche se non dispone dei dati necessari per dare una risposta chiara o per fare una affermazione. In breve, deve poter dare una risposta plausibile a una domanda come: « Che cosa pensa lei di Picasso?

La distinzione fra « capire » e « dare una risposta plausibile » è ben lontana dall'essere chiara e netta: noi cerchiamo di chiarirla con un'analogia fra due stati mentali di cui ogni lettore deve avere avuto esperienza. Da un lato, l'intenso sforzo per comprendere l'essenza di un argomento: dall'altro lo stato mentale di chi non è interessato in una discussione, ma che, per non sembrare scortese, accenna col capo e mormora frasi non compromettenti come: « Molto interessante! ».

È tipico dello stato dell'arte attuale il fatto che i programmi di conversazione finora scritti tendono ad accentuare l'uno o l'altro di questi due aspetti, ma non tutti e due. Siamo ancora nello stadio in cui i ricercatori tentano di mettere a punto una tecnica per volta, anziché introdurne diverse nello stesso programma.

Tra i numerosi programmi scritti negli ultimi anni, due, detti ELIZA e STUDENT, illustrano abbastanza bene i due estremi (rispettivamente « conversazione con scarsa comprensione » e « comprensione con scarsa conversazione »).

ELIZA cerca di tenere una conversazione plausibile senza tentare di « capire » ciò di cui si parla. La parola « capire » è usata in questo senso: se un uomo si comportasse come è descritto qui sotto, noi diremmo certamente che egli rispondeva senza capire l'argomento discusso.

ELIZA è, in sé, un programma di carattere generale che accetta frasi in inglese e risponde con proprie frasi, ottenute dalle frasi in ingresso mediante certe trasformazioni. Queste trasformazioni sono determinate da un « copione », ed ELIZA si comporta in modo assai differente a seconda di quale « copione » sta seguendo. Secondo uno di tali copioni, detto DOCTOR, ELIZA risponde in modo molto simile a quello di uno psichiatra. Esso ha prodotto molte notevoli conversazioni, di cui la seguente è un esempio. (Le risposte di ELIZA sono in maiuscolo)

Gli uomini sono tutti uguali.
IN CHE MODO.
Ci stanno sempre seccando per una cosa o per un'altra.
PUO' PENSARE A UN ESEMPIO SPECIFICO.

Bene, il mio ragazzo mi ha fatto venire qui.
IL SUO RAGAZZO L'HA FATTA VENIRE QUI.
Dice che sono sempre depressa.
MI RINCRESCE SENTIRE CHE LEI È DEPRESSA.
È vero. Io sono infelice.
LEI PENSA CHE IL VENIRE QUI L'AIUTERA' A NON SENTIRSI INFELICE.
Io ho bisogno d'aiuto, questo è certo.
CHE COSA VORREBBE DIRE PER LEI AVERE AIUTO.
Forse potrei imparare ad andare d'accordo con mia madre.
MI DICA QUALCOSA DI PIU' DELLA SUA FAMIGLIA.
Mia madre si prende cura di me.
CHI ALTRI NELLA SUA FAMIGLIA SI PRENDE CURA DI LEI.
Mio padre.
SUO PADRE.
Lei assomiglia in qualche modo a mio padre.
CHE RASSOMIGLIANZA VEDE.
Lei non è molto aggressivo ma penso che non voglia che io me ne accorga.
CHE COSA LE FA PENSARE CHE IO NON SIA MOLTO AGGRESSIVO.
Lei non discute con me.
PERCHE' LEI PENSA CHE IO NON DISCUTA CON LEI.
Lei ha paura di me.
LE FA PIACERE CREDERE CHE IO HO PAURA DI LEI.
Mio padre ha paura di tutti.
COS'ALTRO LE VIENE IN MENTE QUANDO PENSA A SUO PADRE.
Sgridate.
HA QUALCOSA A CHE FARE QUESTO CON IL FATTO CHE IL SUO RAGAZZO L'HA FATTA VENIRE QUI.

* * *

Il metodo seguito da ELIZA per produrre risposte come quelle sopra riportate è descritto dal suo progettista, J. Weizenbaum del MIT, come segue:
« La procedura di massima del programma è molto semplice: il testo è letto ed esaminato per trovare una parola chiave. Se questa è trovata, la frase viene trasformata seguendo una regola associata con la parola chiave: se non, si richiama una osservazione senza contenuto, o, in certe condizioni, una trasformazione precedente. Il testo elaborato o richiamato viene poi stampato ».
Si prenda in esame la frase «Io sono molto infelice in questi giorni ». Supponiamo che uno straniero, con una limitata conoscenza dell'inglese, ma con un buon orecchio abbia sentito pronunciare questa frase ma abbia capito solo le prime due parole «Io sono». Desiderando sembrare interessato, e forse anche comprensivo, egli può rispondere «Da quanto tempo Lei è stato infelice in questi giorni? ». Ciò che egli ha dovuto fare è stato di applicare alla frase originale una specie di maschera, una parte della quale coincide con le due parole « Io sono », e l'altra isola le parole «molto infelice in questi giorni ». Egli deve anche avere un « attrezzo di rimontaggio » associato specificamente a questa maschera, che prescrive che ogni frase della forma « Io sono BLAH » può essere trasformata in: « Da quanto tempo Lei è BLAH », qualunque sia il significato di BLAH. Un esempio un poco più complicato è dato dalla frase « Sembra che tu mi odii ». Ora lo straniero capisce solo le parole « tu » e « mi ». Egli applica la maschera che suddivide la frase nelle quattro parti:
1) Sembra che 2) tu 3) mi 4) odii
di cui egli capisce solo la seconda e la terza parte. la regola di rimontaggio potrebbe allora fornire:
« Che cosa ti fa pensare che io ti odii »
e cioè, può scartare la prima componente, tradurre le due parole note « tu » e « mi » in « io » e « ti », e attaccare una frase di uso generale « Che cosa ti fa pensare » davanti alla frase ricostruita.
Il « copione » DOCTOR consiste di circa una pagina e mezza di regole di scomposizione e ricomposizione come descritte sopra. E' veramente sorprendente che un insieme di regole così relativamente semplice possa produrre una conversazione così credibile come quella riportata. Tuttavia, questa semplicità ha un prezzo. Nel copione discusso, uno degli scopi di ELIZA è quello di nascondere la sua mancanza di comprensione. Se nessuna regola di scomposizione può essere applicata, la risposta può essere una frase come « Parliamo di qualcos'altro. Dimmi qualcosa di questo e di quello ». Una frase come questa, che non richiede alcuna comprensione della discussione precedente, può sempre entrare nella conversazione, purché non troppo spesso.
Il secondo programma in discussione, detto STUDENT, risolve problemi verbali di algebra come i seguenti:
« La somma di tre numeri è 9. Il secondo numero è il doppio del primo numero, più tre. Il terzo numero è uguale alla somma dei primi due numeri.
Trovare i tre numeri ».
« Maria ha l'età doppia di quella che aveva Anna quando Maria aveva l'età che ha Anna ora. Se Maria ha 24 anni, quanti anni ha Anna? ».
Quando gli viene presentato il primo problema, STUDENT stampa:
IL PRIMO NUMERO E' 0,5
IL SECONDO NUMERO E' 4
IL TERZO NUMERO E' 4,5.

Per il secondo problema, esso stampa:
ANNA HA 18 anni.
Per quanto riguarda STUDENT, il mondo consiste di descrizioni verbali di equazioni algebriche, e, in particolare, di equazioni di primo grado. Quando una descrizione di questo genere gli viene presentata, STUDENT procede ad identificare le incognite contenute e le relazioni fra esse, e a tradurre la descrizione verbale in un insieme di equazioni nella notazione matematica normale.
Nel primo problema, STUDENT identifica le tre incognite p (primo numero), s (secondo numero) e t (terzo numero) e poi stabilisce il sistema di equazioni:

$$p + s + t = 9$$
$$s = 2p + 3$$
$$t = p + s$$

Giunto a questo punto, il resto è ovvio, dato che vi sono modi ben noti per risolvere sistemi di equazioni lineari. Quello che ci interessa in STUDENT sta tutto nella sua capacità di tradurre frasi da un ristretto, ma informale, sottoinsieme della lingua inglese, in equa-

NANOCOMPUTER® Z80

Sistema basato sulla CPU Z80 studiato dalla SGS-ATES espressamente per impieghi didattici.

- **IL PIU' POTENTE SISTEMA DIDATTICO SUL MERCATO**
  4K di RAM, 2K di ROM, interfaccia per terminale seriale e cassette magnetiche, 4 porte di I/O, tastiera a 26 tasti, display a 8 digit, accessibilità al bus completa.
- **UTILIZZABILE ANCHE PER SVILUPPO HARDWARE**
  Una scheda addizionale contenente un breadboard senza saldature e dotata di interruttori ed indicatori luminosi, permette di sviluppare circuiti di interfaccia di crescente complessità.
- **MASSIMA FLESSIBILITA' ED ESPANDIBILITA'**
  Espansione sulla scheda fino a 16K di RAM, 8K di ROM, USART, stampante parallela, espansione attraverso schede addizionali fino a 64K di RAM/ROM, interfaccia video e floppy disk.
- **NON SOLO UN MANUALE DI ISTRUZIONE**
  Tre libri in italiano, pensati come parte integrante del sistema.
- **COMPLETO SUPPORTO SOFTWARE E HARDWARE**
  Un monitor da 2K, assembler/editor/debugger, BASIC, tutto su una sola scheda. Kit di espansione, alimentatori, schede per esperimenti, schede a wire wrap, connettori, cavi ...

zioni algebriche espresse in notazione formale. La nostra discussione di come STUDENT può eseguire tale traduzione è necessariamente semplificata, ma speriamo di rendere chiare le idee e le tecniche principali.
STUDENT suddivide le parole e le espressioni nelle frasi che analizza in tre categorie: variabili, operatori e sostituenti.
I sostituenti sono espressioni che STUDENT sostituisce immediatamente con altre. Per esempio, la parola « doppio » è sempre sostituita da « 2 volte ». Lo scopo di tali sostituzioni è quello di ridurre il numero delle forme linguistiche, col sostituire molte diverse e speciali costruzioni con un minor numero di costruzioni più generali.
Gli operatori sono espressioni che denotano le operazioni aritmetiche che STUDENT può compiere (e cioè addizione, sottrazione, moltiplicazione, divisione, elevazione a potenza) o combinazioni di tali operazioni (come la somma di *x* col prodotto di *y* e *z*), e, infine, la relazione di uguaglianza.
Qualcuna delle forme linguistiche che possono essere usate per esprimere le operazioni aritmetiche e la relazione di eguaglianza sono elencate nella tabella:

Sommario delle forme linguis he per esprimere le funzioni aritmetiche e la relaz e di eguaglianza

$x = y$ *è y; x uguaglia y; x è uguale a y.*
$x + y$ x più y; la somma di x ed y; x aggiunto ad y,
$x - y$ x meno y; la differenza fra x ed y; y tolto da x.
$x \cdot y$ x volte y; x moltiplicato y; x y (se x è un numero).
x/y x diviso per y; x per ciascun y.

La terza categoria di parole ed espressioni identificata da STUDENT è quella delle variabili. Una variabile è una sequenza di parole che STUDENT prende come rappresentante di una incognita del problema presentatogli. STUDENT non cerca affatto di comprendere il significato di una variabile: egli non fa altro che identificare quali espressioni sono variabili, e quali frasi indicano la stessa incognita.
Se STUDENT può risolvere un problema verbale in cui si presenta la variabile « l'età di Anna », esso può anche risolvere un problema simile in cui ogni espressione « l'età di Anna » è sostituita da una parola senza senso come « fgh ». Per il fatto che STUDENT non cerca di comprendere alcunchè circa la natura di una data variabile, risulta ovvio che la capacità di STUDENT di riconoscere differenti espressioni che indicano la stessa variabile è assai limitata.
Per esempio, STUDENT non riconoscerebbe che « il numero di anni trascorsi dalla nascita di Anna » indica la stessa incognita che « l'età di Anna ».
Descriviamo ora il metodo principale usato da STUDENT per riconoscere le equazioni algebriche in travestimento verbale. Dopo aver sostituito tutti i sostituenti, STUDENT ricerca se vi sono operatori, confrontando le espressioni delle frasi di ingresso con le espressioni di operatore registrate in un catalogo. Poi identifica come variabili le sequenze di parole comprese fra gli operatori e decide quali sequenze indicano la stessa incognita. (Come si è detto prima, STUDENT riconosce due espressioni come indicanti la stessa incognita solo se esse sono identiche o differiscono di molto poco).
Se le frasi di ingresso sono del tipo che STUDENT può comprendere, a questo punto STUDENT ha costruito un sistema di equazioni algebriche in notazione matematica formale. Se sa risolvere queste equazioni, esso procede alla loro soluzione e stampa i risultati. In caso contrario, per esempio se qualche equazione non è lineare, o se vi sono meno equazioni che incognite, STUDENT domanda: « Conoscete altre relazioni fra queste variabili? ». Se la risposta è «No», esso stampa: « Non so risolvere questo problema ».
E' notevole il fatto che questo modo di affrontare il problema permette a STUDENT di fare con successo quello che fa. Naturalmente, se si è capito il modo di operare di STUDENT è facile costruire esempi che lo fanno sbagliare. Per esempio, in un problema che contenga la variabile « il numero di volte che la vidi », STUDENT identifica subito « volte » come l'operatore di moltiplicazione e ne deduce che vi sono due variabili: « il numero di » e « che la vidi ».
Tuttavia, i problemi che STUDENT ha risolto sembrano indicare che, entro il suo ristretto campo algebrico, esso può competere con lo studente medio delle scuole medie superiori. Ciò dà un certo peso alla conclusione di D.G. Bobrow, che ha scritto STUDENT: « Io penso che siamo ancora lontani dal saper scrivere un programma che possa capire tutta, o almeno in gran parte, la lingua inglese. Tuttavia, nel suo ristretto campo di competenza, STUDENT ha dimostrato che possono essere costruite macchine che "capiscono"».

Può una macchina riprodurre se stessa?

L'autoriproduzione, e cioè la capacità di un organismo di produrne altri simili a sé, è comunemente considerata una prerogativa degli organismi viventi, e un carattere indicativo della distinzione fra materia vivente e materia non vivente. I recenti risultati della sintesi in laboratorio di molecole complesse capaci di riprodursi sono stati indicati come creazione di « vita in provetta », attestando ancora una volta che, comunemente, vita e autoriproduzione sono concetti inseparabili. Tuttavia, il fatto che vi sono, in natura, diversi fenomeni molto simili all'autoriproduzione e relativi a esseri non viventi, (per esempio, l'accrescimento di certi cristalli), suggerisce che la distinzione fra cose che si riproducono e cose che non lo fanno non sia necessariamente la stessa che fra cose viventi e cose non viventi.
Un argomento apparentemente plausibile, secondo cui appare impossibile costruire una macchina autoriproducentesi, è il seguente: se una macchina A costruisce una macchina B, essa deve certamente contenere una descrizione completa di B; ed inoltre A deve comprendere certe altre parti, quali un dispositivo di comando capace di interpretare tale descrizione, organi meccanici capaci di effettuarne la costruzione, ecc. Perciò è chiaro che A deve essere più complessa di B, e che nessuna macchina può costruirne un'altra della stessa complessità.
L'esperienza delle macchine che conosciamo sembra confermare questo argomento. Per esempio, in una linea di montaggio, una macchina che non faccia altro che avvitare a fondo delle viti deve essere sostanzialmente più complessa di una vite. Benchè appaia intuitivamente convincente, tale argomento è

ingannevole, come mostrerà la discussione seguente. Noi abbiamo molta esperienza di macchine relativamente semplici, ma la nostra intuizione di che cosa possano fare, in via di principio, le macchine e soprattutto macchine molto complesse, non è granchè buona. John von Neumann si è occupato di tale materia e l'ha studiata a fondo negli anni '50. Egli ha suggerito che la capacità di un organismo di riprodurre se stesso sia una indicazione della complessità di tale organismo, e si è occupato dello studio degli «automi autoriproducentisi», che possono essere definiti con rigore e completezza, piuttosto che degli organismi viventi, che sono così complessi che nessuno li conosce completamente.
Nel suo studio, von Neumann ha preso in considerazione alcuni differenti modelli di macchine, due dei quali saranno discussi qui: il primo è un dispositivo meccanico, costruito in modo abbastanza tradizionale, con un montaggio di parti elementari; il secondo è un modello algebrico astratto. Il primo modello ha il vantaggio di richiamarsi intuitivamente all'esperienza che abbiamo di dispositivi meccanici, e perciò la costruzione di una macchina autoriproducentesi può essere facilmente capita nelle sue idee essenziali. Ha lo svantaggio che la sua descrizione dettagliata, per esempio sotto forma di disegni costruttivi, sarebbe assai complessa e richiederebbe un tempo eccessivo. Il secondo modello cerca di evitare di preoccuparsi eccessivamente dei particolari e presenta in modo semplificato l'idea dell'autoriproduzione. È allora possibile effettuare un progetto particolareggiato di una macchina autoriproducentesi. Von Neumann ha fatto ciò in un manoscritto di circa 200 pagine. Lo svantaggio del secondo modello, agli scopi espositivi, è che esso richiede un notevole sforzo di pensiero per capire in quale senso precisamente questo modello algebrico contiene la nozione dell'autoriproduzione della macchina. Noi perciò dedicheremo solo poche parole a tale modello, e daremo poi un profilo della costruzione di una macchina autoriproducentesi secondo il primo modello di von Neumann.
Nel modello algebrico, usualmente detto il modello «cellulare» o « a mosaico », lo spazio è rappresentato come uno schieramento infinito bidimensionale e rettangolare di celle identiche, e si ammette che il tempo proceda per intervalli discreti. Ogni cella è essa stessa una semplice macchina astratta, che, ad ogni istante di tempo, può assumere uno stato fra un numero finito di stati, ed è capace di scambiare segnali con le quattro celle adiacenti. Vi è una funzione di transizione che determina il comportamento nel tempo di ogni cella, prescrivendo un cambiamento di stato da un istante al successivo, che dipende solo dallo stato presente della cella e delle sue quattro vicine. Uno degli stati possibili di ogni cella è detto «stato ineccitabile»: una cella in tale stato può considerarsi una cella vuota. Nel modello algebrico una macchina si identifica con una configurazione geometrica finita di celle contigue e con la specificazione dello stato di ogni cella, ed opera come segue. Prima di un istante iniziale t_0, tutte le celle dello schieramento infinito sono nello stato non eccitabile: al tempo t_0 tutte le celle della detta configurazione geometrica sono portate in stati specificati: le altre restano come sono. Da allora in poi, l'operazione è comandata, passo a passo, dalla funzione di transizione, che determina i cambiamenti individuali di stato per ogni cella. Si può allora chiamare autoriproducentesi una macchina (una configurazione di celle contigue in dati stati), se ad un certo tempo finito dopo t_0 vi sono due configurazioni finite contenute nello schieramento infinito, ognuna uguale alla macchina originale.
Descriviamo ora il primo modello di macchina autoriproducentesi, il dispositivo meccanico. Si immagini un grande magazzino contenente una grande, potenzialmente illimitata, provvista di un numero finito di parti meccaniche. Esse sono le « parti elementari » di cui è composta la macchina, e debbono essere di una varietà tale da permettere di costruire con esse veicoli mobili e provvisti di sorgenti autonome di energia. Esse possono essere, per esempio, ruote, ingranaggi, alberi, viti e dadi, nastri di carta, batterie, ecc. Come vedremo, la descrizione precisa di questa provvista di parti può presentare difficoltà, ma, per il momento, supponiamo che una tale raccolta di parti possa essere convenientemente definita. La questione della possibile esistenza di una macchina autoriproducentesi in questo ambiente può essere formulata come segue: dato un inventario sufficientemente ricco di parti elementari, può esistere una macchina composta di tali parti, che possa aggirarsi per questo magazzino, prelevare le parti dagli scaffali, e montarle assieme in modo da costruire una copia di se stessa, e cioè una seconda macchina identica, che possa poi procedere a costruire una terza copia, e così via?
Ora dobbiamo occuparci della precisa descrizione della provvista delle parti. E' abbastanza semplice evitare il problema mediante una improbabile scelta delle parti elementari. Per esempio, la questione diventa banale se una delle parti elementari è definita come l'intera macchina senza la batteria, o senza qualcosa di egualmente semplice: in tale caso l'autoriproduzione consiste semplicemente nella installazione della batteria. Si resta del tutto insoddisfatti da un argomento di questo tipo, perchè esso assume per una parte elementare essenzialmente le stesse proprietà che si cerca di stabilire per la macchina completa. D'altra parte, se le parti elementari si scelgono troppo piccole o troppo semplici, allora le proprietà essenziali della macchina possono venire nascoste dai particolari della costruzione mediante tali parti. Non vi sono quindi regole rigorose, ma solo il senso comune, per suggerire la definizione delle parti elementari: il criterio per stabilire la opportunità di una certa scelta sembra essere quello di stabilire se il problema che ne risulta è o no « interessante ». Lo stesso von Neumann ha descritto, una volta, un gruppo di una dozzina di parti elementari che egli pensava potessero essere usate per costruire la macchina, benchè non ne abbia effettuato la descrizione in termini di tali parti. Egli ha invece suddiviso la macchina autoriproducentesi in tre macchine più piccole, nessuna delle quali è capace di riprodurre se stessa, e che sono tutte macchine plausibili. Queste tre macchine sono: 1) un costruttore universale U; 2) un duplicatore di nastro D; 3) un automa di governo C. Non descriveremo la laboriosa costruzione di tali macchine da una raccolta di parti elementari, ma facciamo appello alla intuizione del lettore che tale costruzione è possibile in via di principio. Descriveremo invece senz'altro queste macchine e il loro funzionamento.

Il costruttore universale U è la parte più critica del modello di von Neumann di una macchina autoriproducentesi. U è una macchina che, data una descrizione (M) di una macchina qualunque, composta delle parti P_1 ...P_k, può fabbricare con tali parti un esemplare della macchina M. L'analogia con un calcolatore universale *(general purpose)* che, data una sufficiente capacità di memoria può essere programmato per calcolare ogni funzione « calcolabile », convincerà la maggior parte di coloro che hanno familiarità con calcolatori che una tale macchina può essere costruita. Un impianto industriale comandato da un calcolatore è provvisto di linee di montaggio è abbastanza simile alla macchina U. Un progetto completo di U, con la specifica della lista delle parti, della descrizione, di un algoritmo di costruzione, appare certamente come un notevole sforzo di ingegneria, ma rientra del tutto nelle capacità della tecnologia moderna.

Il costruttore universale U richiede, per poter funzionare una descrizione non ambigua, in forma leggibile da macchina, della macchina che costruisce.
Ciò solleva un altro problema nella scelta della classe delle parti elementari: e cioè, le parti debbono essere scelte in modo da essere sicuri che esiste una descrizione *finita* di ogni macchina M che può essere costruita con esse.
Ciò non impone in pratica una grande restrizione, poichè è perfettamente ragionevole supporre un insieme di parti tali che, se due parti debbono essere unite in qualche modo, esse possano essere unite solo in un numero finito di modi.
Una volta scelta una conveniente classe di parti, non vi sono speciali difficoltà nel progettare un linguaggio formale che descriva le parti stesse e il modo in cui debbono essere interconnesse. Supporremo che una descrizione (M) di una macchina M sia codificata in forma di una banda di carta perforata, ma qualunque descrizione leggibile da macchina è soddisfacente.
Le altre due macchine componenti sono assai più semplici di U. Il duplicatore di zona D è semplicemente una macchina che legge una banda di carta perforata e ne fa una copia perforando l'identica configurazione di fori su una banda vergine. L'automa di governo, C, è una macchina semplice, che può comandare il funzionamento di U e di D, e cioè avviarle e fermarle, inserirvi zone, ecc. La funzione specifica di C è chiarita dalla descrizione del comportamento della macchina composta U + D + C.

Sia M una macchina qualunque, e (M) la sua descrizione in termini di configurazione P_1 ... P_k, perforate sulla banda di carta. (M) sia unita alla macchina U + D + C. Dapprima la macchina di governo C inserisce la banda (M) nel duplicatore D, ottenendo due copie di (M). Poi C inserisce una copia di (M) nel costruttore universale C, che quindi costruisce la macchina M secondo la descrizione.

Infine, C fornisce l'altra copia di (M) alla macchina M appena fabbricata, e riunisce la descrizione originale (M) a (U + D + C).Così, per una macchina qualunque M, U + D + C + (M) costituisce M + (M). Per l'autoriproduzione, quindi, tutto ciò che si richiede è di fornire alla macchina U + D + C la banda (U + D + C): allora U + D + C + (U + D + C) costruisce precisamente una copia di se stessa. Qui termina la nostra discussione dei modelli di macchine autoriproducentisi di von Neumann. Abbiamo visto una macchina operante in un ambiente interessante, cioè un ambiente in cui le parti elementari possono essere semplici e familiari come quelle che incontriamo nella vita di ogni giorno, eppure tale macchina ha la capacità non comune di autoriprodursi, il che, nella nostra esperienza, abbiamo visto accadere solo negli esseri viventi. Quindi alla domanda posta all'inizio di questo paragrafo si può rispondere affermativamente.

Note e citazioni bibliografiche

L'articolo qui presentato costituisce parte della memoria: G. NIEVERGELT, J.C. FARRAR, *What machines can and cannot do,* Computing Surveys, Vol. 4, No. 2, 1972.

Il falso automa scacchista di Von Kempelen è citato spesso nella letteratura. Si dice che si debba in parte ad esso lo sviluppo di interesse a tale gioco in America nella prima parte del diciannovesimo secolo. La sua interessante storia è sommariamente tracciata in:

R.K. HAGEDORN, *Benjamin Franklin and chess in early America.* Univ. Pennsylvania Press. Philadelphia Pa, 1958.
L'interesse per i programmi di gioco degli scacchi è stato vivo da quando esistono i calcolatori elettronici. Lo sviluppo di tali programmi costituisce un'altra illustrazione del come i calcolatori vengano programmati per eseguire, in modo non banale, una attività che è generalmente considerata di alto livello intellettuale. La prima comunicazione che descrive come possa essere scritto un programma di scacchi è:

C.E. SHANNON, *Programming a digital computer for playing chess,* Philosophy Magazine 41 (March 1950), 356-357. In questo articolo, Shannon accenna anche alla questione dell'intelligenza delle macchine, ed esprime idee simili a quelle di Turing. Questo e diversi altri articoli degni di nota sono riprodotti nella raccolta:

J. R. NEWMAN (Ed.). *The world of mathematics,* vol. 4, Simon and Schuster, New York, 1956.
La descrizione di uno dei recenti programmi di scacchi detto MAC HACK VI, e di alcuni semplici giochi si trova in:

R.D. GREENBLATT, D. E. EASTLAKE, S. D. CROCKER, *The Greenblatt chess program,* in Proc. AFIPS 1967 FJCC vol. 31 AFIPS Press, Montvale, N.J. 801-810.
Fra i programmi di scacchi finora scritti il maggior successo l'ha avuto il programma di Samuel, che ha raggiunto il livello di « Maestro ». E' descritto nei due articoli seguenti:
A. SAMUEL, *Some studies in machine learing using the game of checkers,* IBM J.R.& D 3, 3 (July 1954) 210-229; e

A. SAMUEL, *Some studies in machine learing using the game of checkers.* Il recent progress IBM J. R&D. 11,6 (Nov. 1967) 601-617.

La discussione originale di Turing sulla questione se una macchina può pensare è apparsa in

A.M. TURING, *Computing machinery and intelligence*, Mind 9 (1950) 433-460 ed è ristampato in *The world of mathematics*, citato sopra (pp. 2099-2123). Lo scritto di Turing, che è generalmente considerato un classico della letteratura, fu uno dei primi contributi ad alto livello ad una vivace discussione del problema mente-macchina. Per una buona antologia degli scritti filosofici sul soggetto, che comprende lo scritto di Turing e anche qualche critica ad esso, ed è utile per porre nella giusta prospettiva il problema del pensiero delle macchine, vedi:

R. ANDERSON (Ed.), *Minds and machines*, Prentice Hall Inc. Englewood Cliffs, N.J. 1964.
Una descrizione di diversi importanti programmi atti a rispondere a domande poste in linguaggio naturale si trova in

M. MINSKY (Ed.), *Semantic information processing*, MIT Press, Cambridge Mass. 1968. Il capitolo 3 di tale libro descrive il programma risolutore di problemi algebrici STUDENT, che fu sviluppato in una tesi di laurea al MIT:

D.G. BORROW, *Nature language input for a computer problem-solving system* in *Semantic information processing*, 135-215.
Il programma di conversazione ELIZA è stato descritto per la prima volta in

J. WEIZENBAUM, *ELIZA - a computer program for the study of natural language communication between man and machine*, Comm. ACM 9, 1 (Jan. 1966) 36-45 e più tardi in

J. WEIZENBAUM, *Contextual understanding by computers*, Comm. ACM 1O, 8 (Aug. 1967) 474-480.
La discussione di John Von Neumann sulle macchine autoriproducentisi è parte di uno scritto che espone i suoi concetti su una teoria matematica degli automi. E' apparso sotto il titolo *The general and logical theory of automata in*

L.A. JEFFRESS, Cerebral mechanisms in behavior - the Hixon symposium, John Wiley & Sons, New York, 1951, 1-41. E' anche ristampato nel sopracitato *The world of mathematics* (pp. 2070-2098) e in

A.H. TAUB (Ed.), *John von Neumann - collected works*, vol. V, Macmillan Co., New York, 1963, 288-328. Il modello a mosaico di autoriproduzione menzionato nel testo è descritto particolareggiatamente da von Neumann in un manoscritto che fu incominciato nel 1952 e lasciato incompleto al momento della sua morte. Questo manoscritto fu successivamente completato e curato da A.W. Burks, e pubblicato come parte di

J. VON NEUMANN, *Theory of self-reproducing automata (Edited and completed by Arthur W. Burks) Univ. Illinois Press, Urbana, Ill., 1966. L'attività di ricerca in tale campo continua e un sommario recente si trova in*

A.W. BURKS (Ed.), Essays on cellular automata, Univ. Illinois Press, Urbana, Ill., 1970.

Il mio nome è Eliza

Eliza comprende solo l'inglese, in quanto la sintassi di questa lingua risulta più semplice. Originariamente, il programma è stato creato da Joseph Weizebaum nel linguaggio LISP. Questa versione si basa su una descrizione fatta da Steve North sulla rivista "Creative Computing".

Quando Eliza riceve una frase in entrata, la analizza, cercando, per prima cosa, le parole-chiave, tra quelle scritte nella tabella delle parole-chiave (linee 1050-1100). Lo stadio successivo consiste nel coniugare il resto della frase partendo dalla parola-chiave, in modo che, ad esempio, dalla parola I (io) passa a you (tu), etc.

*Eliza dispone di una serie di frasi di risposta che, con l'aiuto delle parole-chiave, vengono scelte dalla tabella con indirizzo 2530. Nelle frasi di risposta con * alla fine, viene aggiunta la frase di risposta coniugata.*

Risulta relativamente facile ampliare il vocabolario di Eliza e inserire quindi proprie frasi di risposta. A tal fine nelle linee 100 e 110 devono essere registrati i nuovi valori del numero di parole e di frasi. N1 è il numero delle parole-chiave, N2 delle coniugazioni ed N3 delle frasi di risposta. Deve anche essere aggiornata la tabella della linea 2530, nella quale va aggiunto il numero di linea (relativo) delle nuove frasi di risposta e quindi il numero delle nuove frasi di risposta per la relativa parola-chiave.

```
10 REM         E L I Z A
20 REM
30 REM
90 CLEAR 500
100 DIM S(38),R(38),N(38)
110 N1=38 : N2=12 :N3=115
120 RESTORE 2530
130 FOR X=1 TO N1
140 READ S(X) ,L: R(X)=S(X): N(X)=S(X)+L-1
150 NEXT X
160 PRINT "HI, I AM ELIZA TELL ME YOUR PROBLEM"
170 LINE INPUT I$
180 I$=" "+I$+" "
190 REM
200 FOR L=1 TO LEN(I$)
210 IF MID$(I$,L,1)="'" THEN I$=LEFT$(I$,L-1)+RIGHT$(I$,LEN(I$)-L):GOTO 210
220 NEXT L
230 IF I$=P$ THEN PRINT "PLEASE DONT REPEAT YOURSELF":GOTO 170
270 REM
280 RESTORE
290 S=0
300 FOR K=1 TO N1
305 IF S>0 THEN 340
310 READ K$
320 A=INSTR(I$,K$)
330 IF A<>0 THEN  S=K:T=A:F$=K$
340 NEXT K
365 IF S>0 THEN K=S:L=T:GOTO 400
370 K=N1 :GOTO 570
390 REM
400 RESTORE 1200
410 C$=" "+RIGHT$(I$,LEN(I$)-LEN(F$)-L+1
420 FOR X=1 TO N2/2
430 READ S$,R$
440 FOR L=1 TO LEN(C$)
450 IF L+LEN(S$)>LEN(C$) THEN 510
480 IF MID$(C$,L,LEN(S$))<>S$ THEN 510
490 C$=LEFT$(C$,L-1)+R$+RIGHT$(C$,LEN(C$)-L-LEN(S$)+1)
495 L=L+LEN(R$)
500 GOTO 540
510 IF L+LEN(R$)>LEN(C$) THEN 540
520 IF MID$(C$,L,LEN(R$))<>R$ THEN 540
530 C$=LEFT$(C$,L-1)+S$+RIGHT$(C$,LEN(C$)-L-LEN(R$)+1)
540 NEXT L
550 NEXT X
555 IF MID$(C$,2,1)=" "THEN C$=RIGHT$(C$,LEN(C$)-1)
560 REM
570 RESTORE 1300
580 FOR X=1 TO R(K) : READ F$:NEXT X
590 R(K)=R(K)+1 : IF R(K)>N(K) THEN R(K)=S(K)
600 IF INSTR(F$,"*")=0 THEN PRINT F$:P$=I$: GOTO 170
610 A=INSTR(F$,"*")
620 PRINT  LEFT$(F$,A-1);C$;RIGHT$(F$,LEN(F$)-A)
630 GOTO 170
1000 REM
1050 DATA "CAN YOUR","CAN I","YOU ARE","YOURE","I DONT","I FEEL"
1060 DATA "WHY DONT YOU","WHY CANT I","ARE YOU","I CANT","I AM","IM "
1070 DATA "YOU ","I WANT","WHAT","HOW","WHO","WHERE","WHEN","WHY"
1080 DATA "NAME","CAUSE","SORRY","DREAM","HELLO","HI ","MAYBE"
1090 DATA "NO","YOUR","ALWAYS","THINK","ALIKE","YES","FRIEND"
1100 DATA "COMPUTER","ELIZA","MONEY","NOKEYFOUND"
1200 REM
1230 DATA " ARE "," AM ","WERE ,"WAS "," YOU "," I ","YOUR ","MY "
1235 DATA " IVE "," YOUVE "," IM "," YOURE "
1300 REM
```

```
1330 DATA "DON'T YOU BELIEVE THAT I CAN*"
1340 DATA "PERHAPS YOU WOULD LIKE TO BE ABLE TO*"
1350 DATA "YOU WANT ME TO BE ABLE TO*"
1360 DATA "PERHAPS YOU DON'T WANT  TO*"
1365 DATA "DO YOU WANT TO BE ABLE TO*"
1370 DATA "WHAT MAKES YOU THINK I AM*"
1380 DATA "DOES IT PLEASE YOU TO BELIEVE I AM*"
1390 DATA "PERHAPS YOU WOULD LIKE TO BE*"
1400 DATA "DO YOU SOMETIMES WISH YOU WERE*"
1410 DATA "DON'T YOU REALLY*"
1420 DATA "WHY DON'T YOU*"
1430 DATA "DO YOU WISH TO BE ABLE*"
1440 DATA "DOES THAT TROUBLE YOU?
1450 DATA "TELL ME MORE ABOUT SUCH FEELINGS."
1460 DATA "DO YOU OFTEN FEEL*"
1470 DATA "DO YOU ENJOY FEELING*"
1480 DATA "DO YOU REALLY BELIEVE I DON'T*"
1490 DATA "PERHAPS IN GOOD TIME I WILL*"
1500 DATA "DO YOU WANT ME TO*"
1510 DATA "DO YOU THINK YOU SHOULD BE ABLE TO*"
1520 DATA "WHY CAN'T YOU*"
1530 DATA "WHY ARE YOU INTERESTED IN WHETHER OR NOT I AM*"
1540 DATA "WOULD YOU PREFER IF I WHERE NOT*"
1550 DATA "PERHAPS IN YOUR FANTASIES I AM*"
1560 DATA "HOW DO YOU KNOW YOU CAN'T*"
1570 DATA "HAVE YOU TRIED?"
1580 DATA "PERHAPS YOU CAN NOW
1590 DATA "DID YOU COME TO ME BECAUSE YOU ARE*"
1600 DATA "HOW LONG HAVE YOU BEEN*
1610 DATA "DO YOU BELIEVE IT IS NORMAL TO BE*"
1620 DATA "DO YOU ENJOY BEING*
1630 DATA "WE WERE DISCUSSING YOU NOT ME."
1640 DATA "OH, I*"
1650 DATA "YOU'RE NOT REALLY TALKING ABOUT ME, ARE YOU?"
1660 DATA "WHAT WOULD IT MEAN TO YOU IF YOU GOT*"
1670 DATA "WHY DO YOU WANT*"
1680 DATA "SUPPOSE YOU SOON GOT*"
1690 DATA "WHAT IF YOU NEVER GOT*"
1700 DATA "I SOMETIMES ALSO WANT*"
1710 DATA "WHY DO YOU ASK?"
1720 DATA "DOES THAT QUESTION INTEREST YOU?"
1730 DATA "WHAT ANSWER WOULD PLEASE YOU THE MOST?"
1740 DATA "WHAT DO YOU THINK?"
1750 DATA "ARE SUCH QUESTION IN YOUR MIND OFTEN?"
1760 DATA "WHAT IS THAT YOU REALLY WANT TO KNOW?"
1770 DATA "HAVE YOU ASKED ANYONE ELSE?"
1780 DATA "HAVE YOU ASKED SUCH QUESTIONS BEFORE?"
1790 DATA "WHAT ELSE COMES TO MIND WHEN YOU ASK THAT?"
1800 DATA "NAMES DON'T INTEREST ME.
1810 DATA "I DON'T CARE ABOUT NAMES. PLEASE GO ON."
1820 DATA "IS THAT THE REAL REASON?"
1830 DATA "DON'T ANY OTHER REASONS COME TO MIND?"
1840 DATA "DOES THAT REASON EXPLAIN ANYTHING ELSE?"
1850 DATA "WHAT OTHER REASONS MIGHT THERE BE?"
1860 DATA "PLEASE DON'T APOLOGIZE!"
1870 DATA "APOLOGIES ARE NOT NECESSARY."
1880 DATA "WHAT FEELINGS DO YOU HAVE WHEN YOU APOLOGIZE?"
1890 DATA "DON'T BE SO DEFENSIVE!"
1900 DATA "WHAT DOES THAT DREAM SUGGEST TO YOU?"
1910 DATA "DO YOU DREAM OFTEN?"
1920 DATA "WHAT PERSONS APPEAR IN YOUR DREAMS?"
1930 DATA "ARE YOU DISTURBED BY YOUR DREAMS?"
1940 DATA "HOW DO YOU DO ... PLEASE STATE YOUR PROBLEM."
1950 DATA "YOU DON'T SEEM QUITE CERTAIN."
1960 DATA "WHY THE UNCERTAIN TONE?"
1970 DATA "CAN'T YOU BE MORE POSITIVE?"
1980 DATA "YOU AREN'T SURE?"
1990 DATA "DON'T YOU KNOW?"
2000 DATA "WHY NO*"
2010 DATA "DON'T SAY NO ITS ALWAYS SO NEGATIVE "
2020 DATA "WHY NOT?"
2030 DATA "ARE YOU SURE?"
2040 DATA "WHY NO?"
2050 DATA "WHY ARE YOU CONCERNED ABOUT MY*"
2060 DATA "WHAT ABOUT YOUR OWN*"
2070 DATA "CAN'T YOU THINK OF A SPECIFIC EXAMPLE?"
2080 DATA "WHEN?"
2090 DATA "WHAT ARE YOU THINKING OF?"
2100 DATA "REALLY, ALWAYS?"
2110 DATA "DO YOU REALLY THINK SO?"
2120 DATA "BUT YOU ARE NOT SURE YOU*"
2130 DATA "DO YOU DOUBT YOU*"
2140 DATA "IN WHAT WAY?"
2150 DATA "WHAT RESEMBLANCE DO YOU SEE?"
2160 DATA "WHAT DOES THE SIMILARITY SUGGEST TO YOU?"
2170 DATA "WHAT OTHER CONNECTIONS DO YOU SEE?"
2180 DATA "COULD THERE REALLY BE SOME CONNECTIONS?"
2190 DATA "HOW?"
2200 DATA "YOU SEEM QUITE POSITIVE."
2210 DATA "ARE YOU SURE?"
2220 DATA "I SEE."
2230 DATA "I UNDERSTAND."
2240 DATA "WHY DO YOU BRING UP THE TOPIC OF FRIENDS?"
2250 DATA "DO YOUR FRIENDS WORRY YOU?"
2260 DATA "DO YOUR FRIENDS PICK ON YOU?"
2270 DATA "ARE YOU SURE YOU HAVE ANY FRIENDS?"
2280 DATA "DO YOU IMPOSE ON YOUR FRIENDS?"
2290 DATA "PERHAPS YOUR LOVE FOR FRIENDS WORRIES YOU."
2300 DATA "DO COMPUTERS WORRY YOU?"
2310 DATA "ARE YOU TALKING ABOUT ME IN PARTICULAR?"
2320 DATA "ARE YOU FRIGHTENED BY MACHINES?"
2330 DATA "WHY DO YOU MENTION COMPUTERS?"
2340 DATA "WHAT DO YOU THINK MACHINES HAVE TO DO WITH YOUR PROBLEM?"
2350 DATA "DON'T YOU THINK COMPUTERS CAN HELP PEOPLE?"
2360 DATA "WHAT IS IT ABOUT MACHINES THAT WORRIES YOU?"
2370 DATA "SAY, DO YOU HAVE ANY PSYCHOLOGICAL PROBLEMS?"
2380 DATA "WHAT DOES THAT SUGGEST TO YOU?"
2390 DATA "I SEE"
2400 DATA "I'M NOT SURE I UNDERSTAND YOU FULLY"
2410 DATA "COME COME ELUCIDATE YOUR THOUGHTS"
2420 DATA "CAN YOU ELABORATE ON THAT ?"
2430 DATA "THAT IS QUITE INTERESTING."
2440 DATA "WHY DO YOU HAVE PROBLEMS WITH MONEY?"
2450 DATA "DO YOU THINK MONEY IS EVERYTHING?"
2460 DATA "ARE YOU SURE THAT MONEY IS THE PROBLEM?"
2470 DATA "I THINK WE WANT TO TALK ABOUT YOU, NOT ABOUT ME "
2480 DATA "WHAT'S ABOUT ME?"
2490 DATA "WHY DO YOU ALWAYS BRING UP MY NAME?"
2530 DATA 1,3,4,2,6,4,6,4,10,4,14,3,17,3,20,2,22,3,25,3
2540 DATA 28,4,28,4,32,3,35,5,40,9,40,9,40,9,40,9,40,9,40,9
2550 DATA 49,2,51,4,55,4,59,4,63,1,63,1,64,5,69,5,74,2,76,4
2560 DATA 80,3,83,7,90,3,93,6,99,7,116,3,113,3,106,7
2570 END
```

TRIBUNA

Introduzione alla nuova tecnologia "VLSI" e ai problemi di interfaccia con il cliente

L'industria dei semiconduttori è emersa in tutto il mondo come l'industria chiave non soltanto per la sua espansione ed importanza, ma perché è alla base di tutta l'industria elettronica.

Recentemente si è fatto un gran parlare circa la nuova generazione tecnologica chiamata "VLSI", o Very Large Scale Integration.
Il "VLSI" fornirà all'uomo un più elevato livello nella creazione di soluzioni elettroniche ai suoi problemi, compreso l'uso di macchine intelligenti.
È evidente che siamo allo stadio iniziale di uno dei più profondi cambiamenti tecnologici di questo secolo: l'abilità di estendere la capacità intellettuale dell'uomo con l'elettronica.

Questa rivoluzione è dovuta a diversi fattori che hanno caratterizzato l'industria dei semiconduttori fin dall'inizio.
Il primo di questi è il fattore riguardante la capacità di realizzare geometrie sempre più piccole.

Trends tecnologici dell'industria dei semiconduttori

L'abilità nel costruire dispositivi a semiconduttore con geometrie sempre più ridotte è stata la forza trainante dietro la crescita dell'industria dei semiconduttori.

Non c'é legge fisica o barriera tecnologica contro questa tendenza almeno per il prossimo decennio.

Attualmente i prodotti più avanzati sono costruiti con tecnologia di 3 micron approssimativamente, che diminuirà fino a tecnologia del submicron verso la fine degli anni 80, prevedendo di far uso della litografia avanzata e implementare la produzione con controllo computerizzato dei processi.

Con il ridursi delle dimensioni geometriche aumenta la capacità delle prestazioni di un dispositivo realizzato in un singolo chip, in quanto possiamo integrare nella stessa area di silicio un numero più elevato di elementi attivi (AEG) (Active Element Group).

Lo stato tecnologico attuale permette di costruire prodotti sempre più avanzati come la 64K RAM dinamica con una densità, includendo le funzioni di controllo, di circa 10^5 AEGs/chip.

Con la tecnologia sub-micron, complessità come 10^6 AEGs per chip saranno possibili verso la fine degli anni 80. Il percorso, per quanto riguarda lo sviluppo delle

memorie nel prossimo decennio, è quindi ben caratterizzato e prevedibile: maggiore complessità e aumentata velocità.

Il futuro dei componenti a semiconduttori

L'aumento della complessità a minor costo per AEG ha portato ad un aumento continuo della domanda di semiconduttori.

Il contenuto di semiconduttori, inteso come percentuale del valore di mercato di apparecchiature elettroniche, è raddoppiato nel decennio passato fino ad arrivare ad un livello del 7%.

Noi crediamo che questo livello aumenterà oltre il 10% alla fine del prossimo decennio.

La tecnologia del semiconduttore continuerà a seguire la storica "learning curve" con i nuovi prodotti VLSI lungo tutto il prossimo decennio. Avremo sistemi programmabili realizzati in un solo componente, con la standardizzazione necessaria per permettere il più vasto uso di circuiti logici VLSI. La possibilità che vengano programmati permetterà la realizzazione a costi competitivi di un largo numero di specifiche applicazioni sistemistiche.

Si può fin d'ora prevedere che sia il produttore di componenti che il produttore di sistemi finiti beneficeranno sia della riduzione dei costi dei componenti in funzione della "learning curve" sia dei ridotti costi di programmazione. Le attività di progettazione di semiconduttori si espanderanno notevolmente fino a prevedere un completo ed efficace sistema di interfaccia realizzato mediante computer per rendere quanto più agevole possibile ogni scambio di informazioni tra il produttore di componenti e quello di sistemi finiti.

Il mercato dei semiconduttori

Basandoci sulle future opportunità e le tendenze dell'industria dei semiconduttori, possiamo provvedere che il mercato dei semiconduttori crescerà dai 9.6 bilioni di dollari del 1979 ad oltre 30 bilioni di dollari alla fine degli anni 80. In particolare, la parte di componenti programmabili crescerà ad un tasso annuale di circa il 26%, raggiungendo i 15 bilioni di dollari alla fine degli anni 80, rappresentando circa la metà del totale mercato dei semiconduttori. Tale crescita non troverà limite nella opportunità di mercato.

L'unico limite prevedibile è l'efficacia con cui l'industria dei semiconduttori riuscirà a preparare e collaborare con la propria clientela, sviluppando un adeguato complesso di interazioni, che faciliterà il processo di trasferimento sul mercato, nei tempi e nei modi adeguati, dei vantaggi offerti dai nuovi traguardi tecnologici raggiunti.

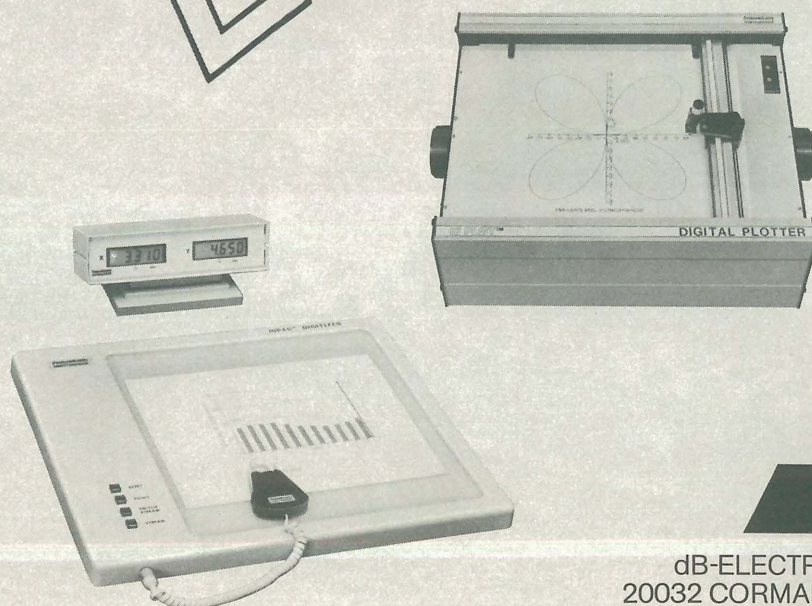
Crescita del mercato dell'elettronica

I continui progressi tecnologici realizzati dall'industria dei semiconduttori avranno un formidabile impatto sul mercato mondiale dell'elettronica durante il prossimo decennio. Il tasso storico di crescita dell'8,3% aumenterà minimo fino all'11,5%, e forse fino al 13,5%.

Noi proiettiamo la crescita del mercato mondiale dell'elettronica a 325 bilioni di dollari alla fine degli anni 80, con un contenuto di semiconduttori di circa il 10%. Se poi teniamo presente il pieno impatto dei nuovi circuiti VLSI, il mercato potrà crescere fino a 400 bilioni di dollari. Noi riteniamo che con la programmazione allo stato solido la tecnologia VLSI sarà quanto prima una realtà dell'industria dei semiconduttori.

Molto dipenderà, per quanto concerne l'uso ed il beneficio che il mondo dell'elettronica riuscirà a ricevere, dalla abilità di produttori di componenti e produttori di sistemi nell'instaurare un nuovo e più adeguato rapporto per far fronte alle esaltanti sfide che le nuove tecnologie pongono ad entrambi.

Texas Instruments

MICROPERIFERICHE
A MICROPREZZI
PER MICROCALCOLATORI
E MICROPROCESSORI
houston instrument
DIGITAL PLOTTER
dB
dB-ELECTRONIC INSTRUMENTS s.r.l.
20032 CORMANO (MI) UFFICI: VIA TORINO, 5
Tel. (02) 6132313-6197201

Il picocomputer - Scheda di unità centrale

Parte II di **D. Del Corso**

La presentazione del "picocomputer" continua con questo articolo che descrive la scheda base di unità centrale. Questa contiene anche ROM, RAM e interfacce di I/O. Insieme al picoperiferico descritto nell'articolo precedente forma un sistema minimo già in grado di eseguire programmi preparati dall'operatore, e nello stesso tempo può essere nucleo base di configurazioni più potenti, da realizzare utilizzando schede di memoria ed interfacce che saranno descritte in seguito.
L'articolo comprende la descrizione completa della scheda di unità centrale ed un breve inserto sullo standard MUBUS, utilizzato per i moduli descritti in questa serie di articoli.
L'argomento trattato in questo articolo sarà completato nel prossimo numero.

a cura del Gruppo Amatoriale Micro-Elaboratori, Torino *e del Laboratorio Didattico di Elettronica Applicata dell'Istituto di Elettronica e Telecomunicazioni del Politecnico di Torino.*

Considerazioni generali

Prima di descrivere nei dettagli i circuiti di questa unità centrale è utile premettere alcune notizie sul "progetto sistemistico".
In altre parole, definire esattamente la struttura dell'oggetto che intendiamo realizzare e quali sono i vincoli da rispettare comunque. La stesura di specifiche

complete e corrette è già buona garanzia per la riuscita di un progetto ed è comunque indispensabile per la documentazione dello stesso.
Dato per scontato che ormai non ha più senso progettare una scheda di sola CPU, occorre decidere cosa altro inserire sullo stesso stampato: memoria, periferici, interfaccia TV....? Esistono essenzialmente due alternative:
- realizzare "piastroni" contenenti tutto o quasi tutto, (anche una tastiera), che nella configurazione minima sono equipaggiati con i soli componenti necessari,
- realizzare moduli base più piccoli, specializzati su funzioni particolari (RAM, ROM, I/O ecc).

La prima soluzione è senz'altro più economica ed affidabile per sistemi a configurazione fissa o comunque rigidamente definita.
La seconda è più "modulare" e permette di calibrare esattamente il sistema sulle necessità dell'utente.
Nel nostro caso inoltre si sceglie a priori di usare lo standard hardware MUBUS (vedi inserto), sia per compatibilità con progetti già esistenti, sia perchè questa scelta semplifica l'interfacciamento di memoria e controllori di periferici di diversi costruttori.
Questo vincola le dimensioni fisiche dei moduli al formato Eurocard (100 X 160 mm), condizionandone anche le "dimensioni funzionali". Lo spazio disponibile su una scheda di queste dimensioni può sembrare poco, ma teniamo conto che, per esempio, con i chip 16K X 1 possiamo porre su una singola eurocard 64 Kbyte X 8 di RAM, e che questo articolo presenta un sistema autonomo con ROM, RAM e I/O e interfacce verso il BUS, tutto realizzato su un'unica scheda. Inoltre l'evoluzione delle tecniche di integrazione concentra determinate funzioni in un numero sempre più ridotto di circuiti integrati; utilizzando RAM 64K X 1 la scheda qui descritta può ospitare (eliminando i buffer e modificando lo stampato) 64 Kbyte di memoria. Va comunque riconosciuto che sulla nostra scheda può trovar posto un numero limitato di componenti.
Questo richiede un progetto circuitale particolarmente oculato e comporta la rinuncia ad alcune comode funzioni quali ad esempio la rilocabilità della memoria locale, l'indirizzamento separato (I/O mapped) dei periferici locali, un restart automatico ad un monitor allocato a qualunque indirizzo, un real time clock e altro. Si è preferito invece lasciare sullo stampato dei posti integrato e zone millefori libere, disponibili per inserire altre funzioni e personalizzare così la scheda sulle necessità specifiche dell'utente. Un elenco dei possibili usi per questo spazio libero è in coda a questo articolo.
Le dimensioni della RAM e ROM locali sono state

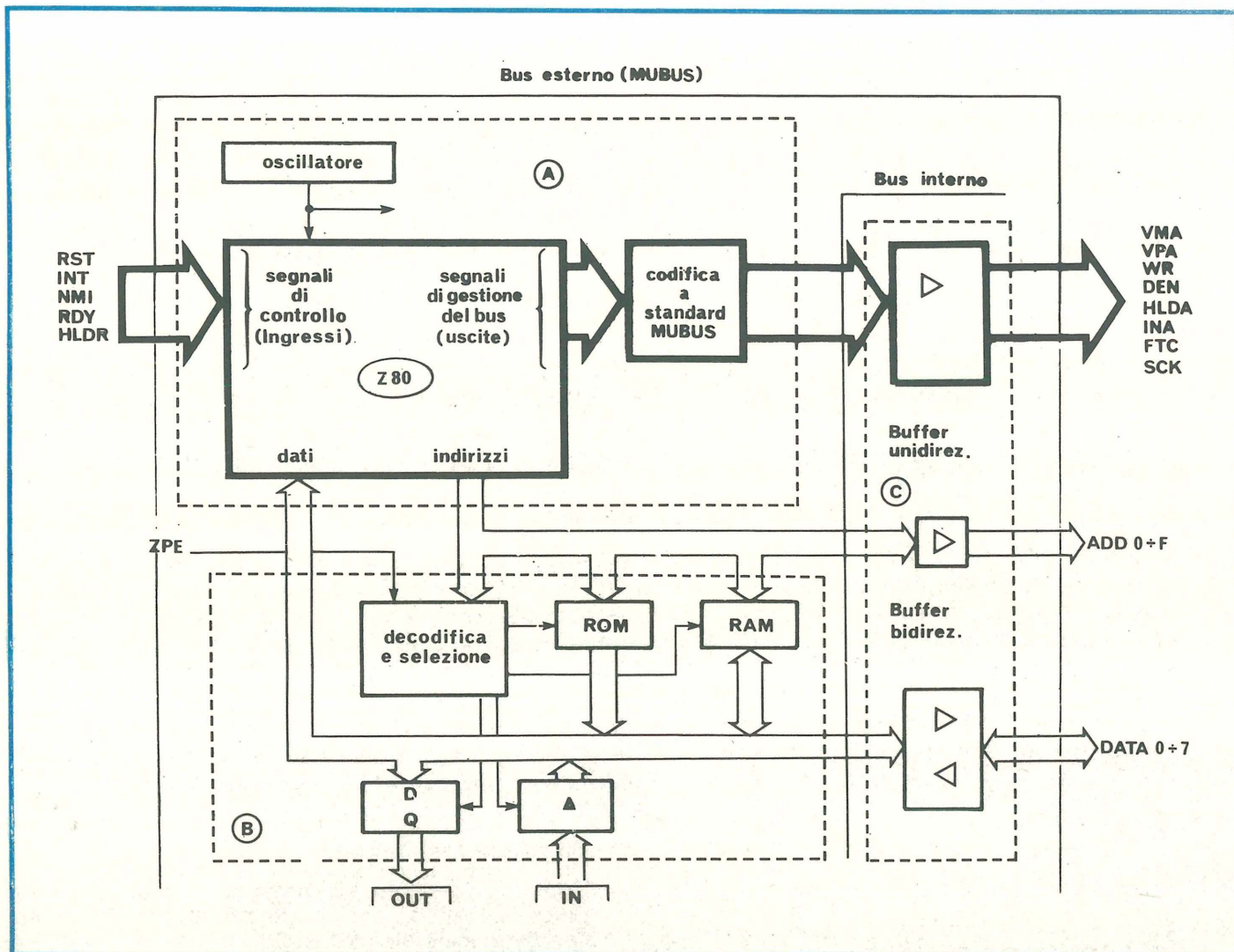



Figura 1 - Schema a blocchi

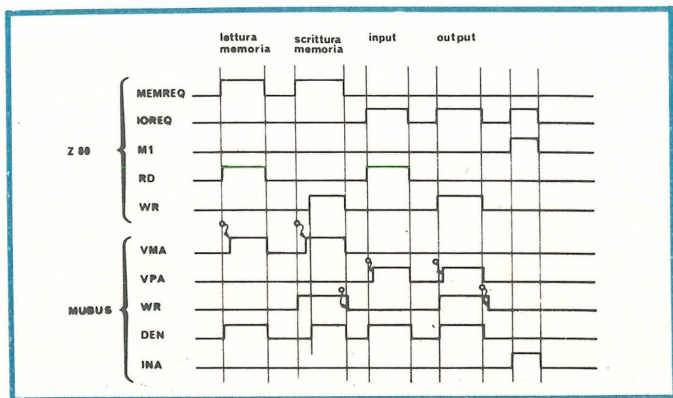



Figura 2 - Codifica Z80 MUBUS per i cicli di trasferimento dati.

scelte in modo da utilizzare componenti attualmente disponibili; si hanno quindi:
- fino a 2 Kbyte di ROM/EPROM, con possibilità di usare 2708, 2716 o PROM più piccole;
- 1 Kbyte di RAM (2 X 2114);
- due registri da 8 bit, mappati su memoria per utilizzare lo stesso decodificatore di indirizzo della ROM e RAM, utilizzabili per operazioni di I/O dall'esterno;
- due uscite impulsive, corrispondenti ad indirizzi non utilizzati sulla piastra, atte a comandare dispositivi esterni con memoria (contatori, registri,...).

Il microprocessore usato è lo Z 80, dotato di un set di istruzioni potente ed esteso e soprattutto in grado di eseguire tutti i vari programmi già disponibili sul mercato per l'8080.
In definitiva, la scheda comprende quindi, oltre al microprocessore con i relativi circuiti di controllo, anche memoria ed interfacce che permettono di utilizzarla sia da sola, come sistema minimo, sia come unità base di sistemi più complessi.

Descrizione funzionale

I circuiti integrati MOS, quali microprocessori e memorie, hanno generalmente un Fan-out di 1 TTL standard, sufficiente per pilotare direttamente piccoli sistemi, ed in particolare tutte le risorse di una scheda singola di medie dimensioni. Invece, per collegare più moduli in vista di una effettiva espandibilità del sistema, occorre interporre sulle linee dei dati, indirizzo e controlli dei circuiti per aumentare la corrente disponibile (buffer) onde poter pilotare tutti gli ingressi collegati a ciascuna linea e le piste della piastra madre. Quindi, nel progettare una scheda che deve funzionare sia autonomamente che insieme ad altre, occorre decidere se bufferizzare o no le diverse linee di bus.

Nel nostro caso la piastra CPU è organizzata su due livelli di bus: bus interno, non bufferizzato, al quale sono collegate tutte le risorse locali quali RAM, ROM e interfacce, e bus esterno, connesso al resto del sistema. I due livelli sono collegati tramite buffer TTL. In questo modo la scheda può funzionare da sola in modo autonomo senza i buffer che devono essere inseriti solo in caso di espansione (i posti integrati sono già disponibili sullo stampato).
Lo schema a blocchi completo della scheda è in fig 1. In esso possiamo individuare tre moduli funzionali:
A - unità centrale e circuiti di gestione e controllo;
B - risorse locali (RAM, ROM, I/O) con i relativi circuiti di selezione;
C - buffer verso il bus esterno.
Il blocco A è il nucleo principale e può operare anche in assenza di B e C. Esso comprende:
- un oscillatore a quarzo,
- la CPU Z 80,
- i circuiti per la codifica dei segnali di gestione secondo lo standard MUBUS.

Il segnale dell'oscillatore viene inviato anche al resto del sistema (segnale SCK, linea 16 B) e la sua frequenza (2.4576 MHz) è quella standard dei generatori di cadenza per trasmissione seriale. È anche possibile usare frequenze diverse (fino a 4 MHz per lo Z 80A); naturalmente, nel caso di frequenze più alte, occorre scegliere memorie con tempo di accesso adeguato. L'oscillatore può funzionare anche con una capacità di 100÷200 pF al posto del quarzo; in tal caso la frequenza va verificata e tarata agendo sul valore della capacità. Questa soluzione è comunque sconsigliata perchè non dà nessuna garanzia di stabilità.

I segnali di controllo del bus Z 80 sono trascodificati allo standard MUBUS attraverso un gruppo di porte e decodificatori.
La temporizzazione per i cicli di accesso alla memoria ed ai periferici è riportata in fig 2. Tutte le informazioni di stato (ciclo di rinfresco, riconoscimento dell'interruzione, accesso a memoria o periferico) sono decodificate ed inviate su linee separate del bus, definite nella tabella dell'inserto sul MUBUS.
Viene anche generato un segnale di scrittura anticipato (WR), che semplifica l'interfacciamento al bus di qualsiasi tipo di memoria e può comandare direttamente la direzione dei buffer sulle linee dati.
Il LED LD segnala l'esecuzione di una istruzione di HALT.
Le varie linee che portano comandi alla CPU (interruzione, reset ecc.),sono direttamente collegate ai rispettivi ingressi dello Z 80, con una resistenza di pull-up per garantire il livello "1" logico (comando non attivo) a riposo.
I diodi ed il condensatore C 4 sulla linea di RESET sono necessari per generare un impulso di RST automaticamente all'accensione (C 4 si carica lentamente verso il + 5V), mantenendo la possibilità di accettare impulsi di reset brevi (il diodo D 2 isola C4 carico dalla linea RST).
Le risorse locali (blocco B in fig 2) sono tutte allocate in un banco di 4 Kbyte, a partire da pagina 0. La ROM inizia allo indirizzo 0000_H, e può essere usata per programmi di monitor o di inizializzazione, da attivare in seguito ad un reset. La mappa completa degli indirizzi è in fig 3.
Il decodificatore di banco è un AND a 4 ingressi negati, che riconosce la configurazione 0000 sui bit di indirizzo ADD C÷ F, realizzato con una connessione wired-and tra uscite a collettore aperto (integrato 74LS05). La selezione della memoria locale viene bloccata dal segnale ZPE (Zero Page Enable), che corrisponde alla linea 24B di bus e può essere controllato da un interruttore interno o esterno alla piastra, o da un bit di un periferico.
Disabilitando le risorse locali della piastra CPU è possibile inserire memoria esterna, RAM o ROM, all'indirizzo 0000_H.
Il decodificatore di banco genera il segnale OCA (On

Ingressi del decodificatore:	C	B	A		
linee di bus:	ADD B	ADD A	$\overline{WR}$	Risorsa	Indirizzo (Hex)
	0	0	0	PUL 1	0000 - 03FF
	0	0	1	ROM	0000 - 03FF
	0	1	0	PUL 2	0400 - 07FF
$\overline{ADD}$ F,E,D,C = 0	0	1	1	ROM	0400 - 07FF
$\overline{VMA}$ = 0	1	0	0	RAM	0800 - 0BFF
$\overline{DEN}$ = 1	1	0	1	RAM	0800 - 0BFF
$\overline{ZPE}$ = 0	1	1	0	OUT	0C00 - 0FFF
	1	1	1	INP	0C00 - 0FFF

Figura 3 - Tabella degli indirizzi di risorse locali.

Card Address), che indica la selezione di una risorsa locale. Questo segnale viene usato come abilitazione (ingresso D) per un decodificatore 3-8 con uscite a collettore aperto (74LS145). La selezione segue la tabella di fig 3; le uscite che selezionano lo stesso dispositivo sono collegate in wired-or.

I periferici sono mappati su indirizzi di memoria per evitare una logica di selezione separata; i registri di ingresso e di uscita corrispondono quindi ad un blocco di indirizzi di memoria, come specificato nella tabella. Eseguendo operazioni di scrittura sugli indirizzi assegnati alla ROM, questa non viene selezionata e si attivano in sua vece le uscite impulsive PUL 1 e PUL 2.

Tutte le linee di I/O sono portate su due zoccoli a 16 piedini (da usare con connettori per cavo piatto), secondo la tabella di fig 4. Ingressi ed uscite sono mescolati perchè il connettore A è previsto per il collegamento con il picoperiferico descritto nell'articolo precedente di questa serie.

I segnali PUL corrispondono ai transistori di uscita a collettore aperto del decodificatore 74LS145, in grado di assorbire 80 mA e di reggere una tensione di 15 V.

Il blocco C comprende semplicemente dei buffer a tre stati che collegano il bus interno al bus esterno, accessibile al connettore della scheda. I buffer sono

Connettore A (□)

Pin		Pin	
1	OUT 4	9	OUT 3
2	OUT 5	10	IN 1
3	OUT 6	11	IN 0
4	OUT 7	12	$\overline{RST}$
5	OUT 0	13	IN 5
6	OUT 1	14	IN 6
7	OUT 2	15	IN 7
8	GND	16	+ 5 V

Connettore B (O)

Pin		Pin	
1	IN 0	9	IN 7
2	IN 1	10	OUT 4
3	IN 2	11	OUT 5
4	IN 3	12	OUT 6
5	IN 4	13	OUT 7
6	IN 5	14	PUL 1
7	IN 6	15	PUL 2
8	GND	16	+ 5 V

Connettore A: **8 linee OUT, 5 linee IN, Reset**

Connettore B: **4 linee OUT, 8 linee IN, 2 PUL**

Figura 4 - Tabella connessioni di I/O

unidirezionali per le linee di indirizzo e di controllo, bidirezionali per le linee dati. L'interfacciamento tra i due livelli di bus mantiene la funzione di HOLD; questo comando porta in TRI-STATE tutte le uscite verso il bus ed il controllo può passare ad un altro master, per esempio un controllore di accesso diretto alla memoria. Esiste anche un comando INH (INHIbit, linea 17 B), che isola il bus esterno da quello in-

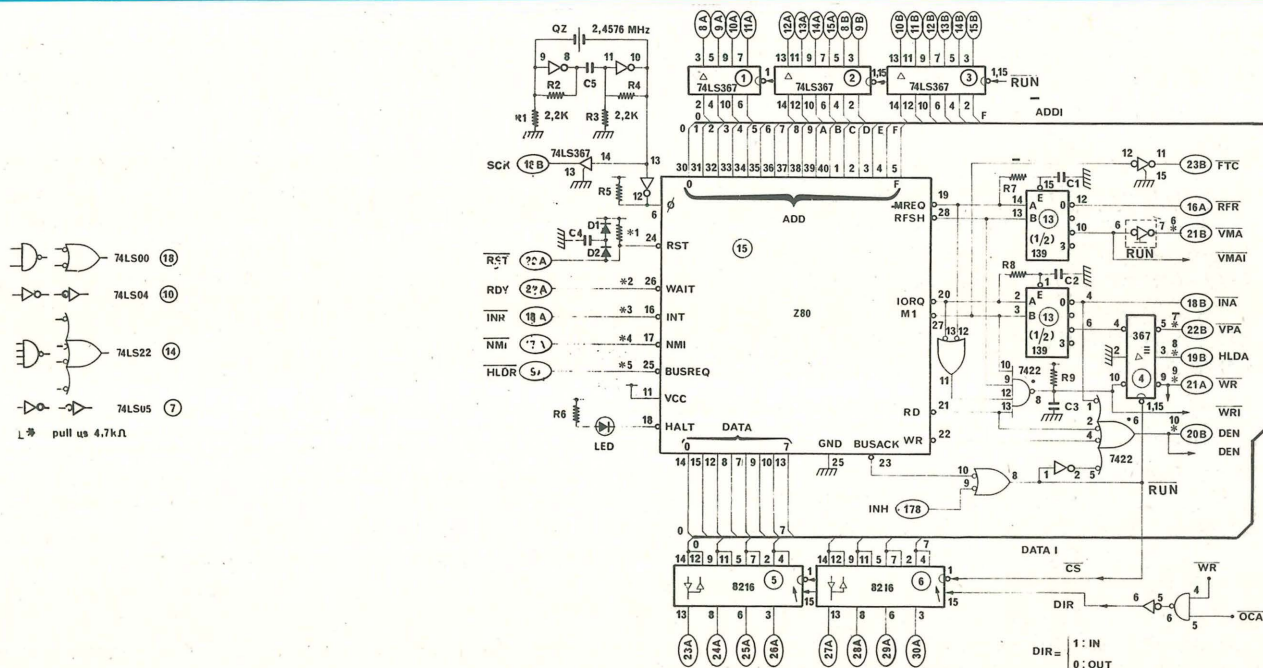
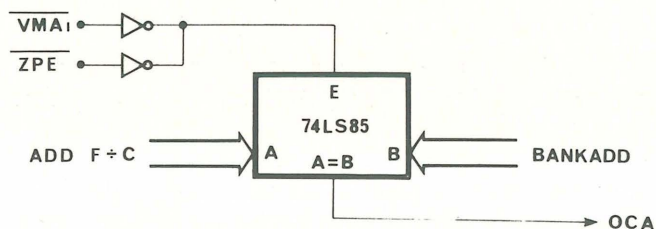
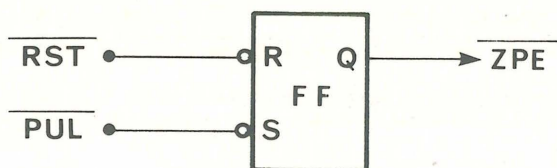
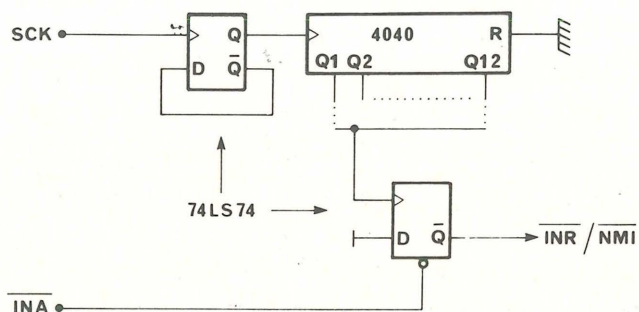



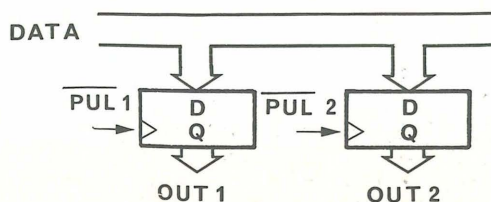
Figura 5 - Schema elettrico completo

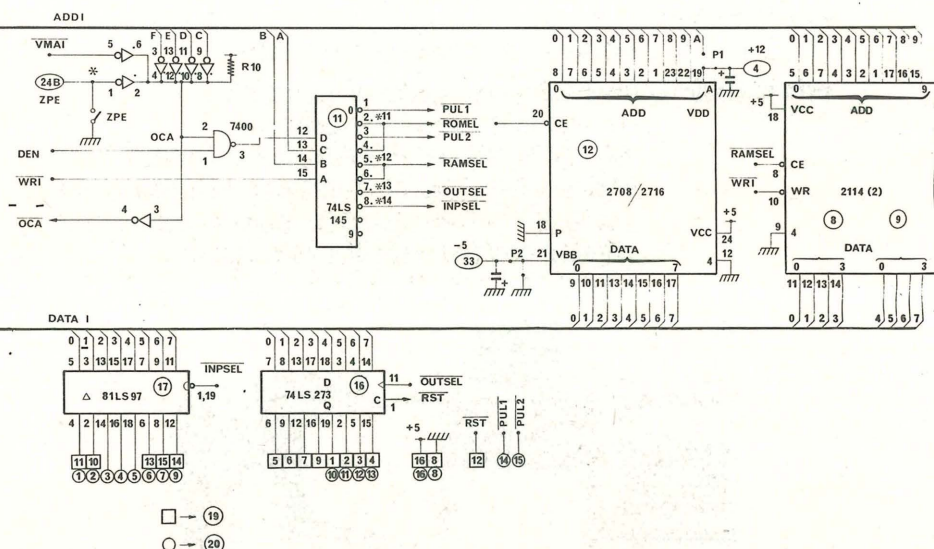
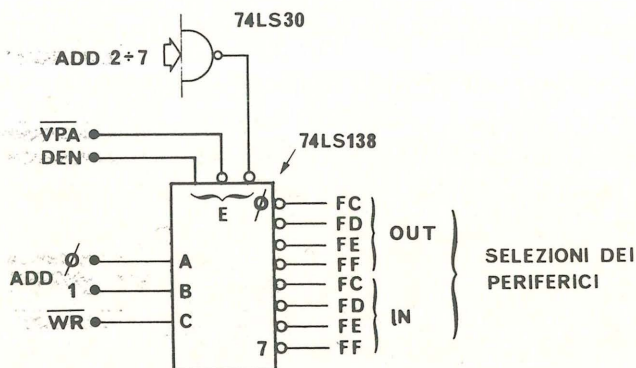
terno, indipendentemente dallo stato del processore.
Questo permette di realizzare sistemi con più unità centrali, che possono lavorare contemporaneamente, ciascuna sulle proprie risorse interne, ed una alla volta sulla memoria e periferici collegati al bus esterno.
Lo schema elettrico completo della scheda è in fig 5.

Possibile uso dei posti integrato liberi

a) Rilocatore per le risorse locali:
Sostituire il decodificatore di banco 0 (74LS05) con un comparatore (p. es. 74LS85).

È ora possibile assegnare la memoria e l'I/O locali ad un qualsiasi banco di 4 Kbyte, presettando opportunamente gli ingressi BANKADD.

b) Bootstrap "fantasma":
La linea ZPE è controllata da un FF set-reset; dopo la esecuzione di un programma di inizializzazione contenuto nella ROM locale viene attivata un'uscita PUL che, attraverso il FF, disabilita le risorse locali ed abilita una RAM esterna da 4 K all'indirizzo 0000_H.

c) Real-time clock:
Aggiungere un divisore ed un doppio FF tipo D.

La presa sul 4040 seleziona la cadenza delle interruzioni (fino a 3 ms circa). Non è prevista vettorizzazione; usare NMI o il modo 0.

- Altre linee di uscita:
È sufficiente aggiungere registri strobati da PUL 1 e PUL 2:

Non è possibile aggiungere linee di ingresso perché non sono più disponibili indirizzi in lettura.

- Mappamento su I/O dei periferici locali:
Sostituire INPSEL e OUTSEL dei periferici locali con

i segnali generati dal circuito qui indicato. Gli indirizzi selezionati vanno da FC a FF_H.

Comunicato

I circuiti stampati delle schede descritte in questo articolo sono disponibili.
I lettori interessati potranno ricevere ulteriori informazioni scrivendo alla redazione di Bit.

ERRATA CORRIGE

Nell'articolo "volete farvi un picocomputer?", pubblicato sul numero scorso, il listing del programma manca dell'inizializzazione dello stack. Inserire all'inizio l'istruzione LD SP,08E 0H.

Lo standard MUBUS

Questo standard hardware definisce un bus per sistemi a microprocessore indipendente dal tipo di unità centrale utilizzato.
I segnali del bus sono tali da permettere di interfacciare molto semplicemente memorie e controllori di periferici di ogni tipo.

Lo standard è nato dal confluire di iniziative di unificazione sorte in diversi laboratori universitari ed industriali, fin dal 1975 (vedi in proposito EUROMICRO NEWSLETTER n° 3,1975 e sgg.).Le specifiche complete del MUBUS sono riportate in "MUBUS STANDARD"Microscope special issue, n°8/77 (editore J.D. Nicoud, p.o.box 141 CH 1007-Lausanne). Una descrizione del MUBUS (con lievi modifiche rispetto allo standard) è comparsa in Elettronica Oggi, n° 2,1979, dal titolo "Una proposta di BUS standard per microcalcolatori modulari" di Bisani - Mezzalira - Negrini.

Il MUBUS è attualmente usato correntemente in parecchie università; sistemi e moduli per uso didattico ed industriale sono prodotti da alcune ditte italiane ed estere.

Lo standard MUBUS comprende una definizione funzionale, che specifica quali comandi esistono e come devono essere usati, ed uno standard fisico (formato delle schede, connettore ecc).

Le tabelle di questo inserto riportano in forma compatta la definizione delle varie linee e l'assegnazione delle connessioni per cartelle formato Europ singolo con connettore diretto 2 X 37.Altri connettori e formati di scheda sono descritti nei riferimenti citati.

Definizione funzionale delle linee di bus

Le linee sono raggruppate per classi di funzioni; in ultima colonna compare il nome usato nei riferimenti citati.

- Controllo dei trasferimenti dati:

ADD	(ADDress)	linee di indirizzo	(ADD)
DATA		linee dati	(DATA)
VMA	(Valid Memory Address)	validazione di indirizzo per la memoria	(ADMEM)
VPA	(Valid Peripheral Address)	validazione di indirizzo per un periferico	(ADPER)
WR	(WRite)	comando di scrittura	(WRITE)
DEN	(Data ENable)	consenso al trasferimento dati	(NODA)
RDY	(ReaDY)	richiesta di attesa (WAIT)	(NOTYET)

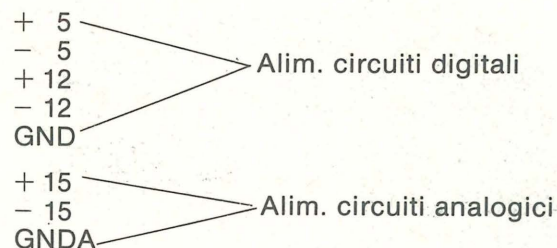
- Comandi al master:

RST	(ReSeT)	inizializzazione	(RESET)
INR	(INteRrupt)	richiesta di interruzione	(INTREQ)
NMI	(NonMaskable Interrupt)	richiesta di interruzione non maschera-bile	
HLDR	(HoLD Request)	richiesta di sospensione (HOLD)	(HOLDREQ)
INH	(INHibit)	isola il master dal bus	(INHIBIT)

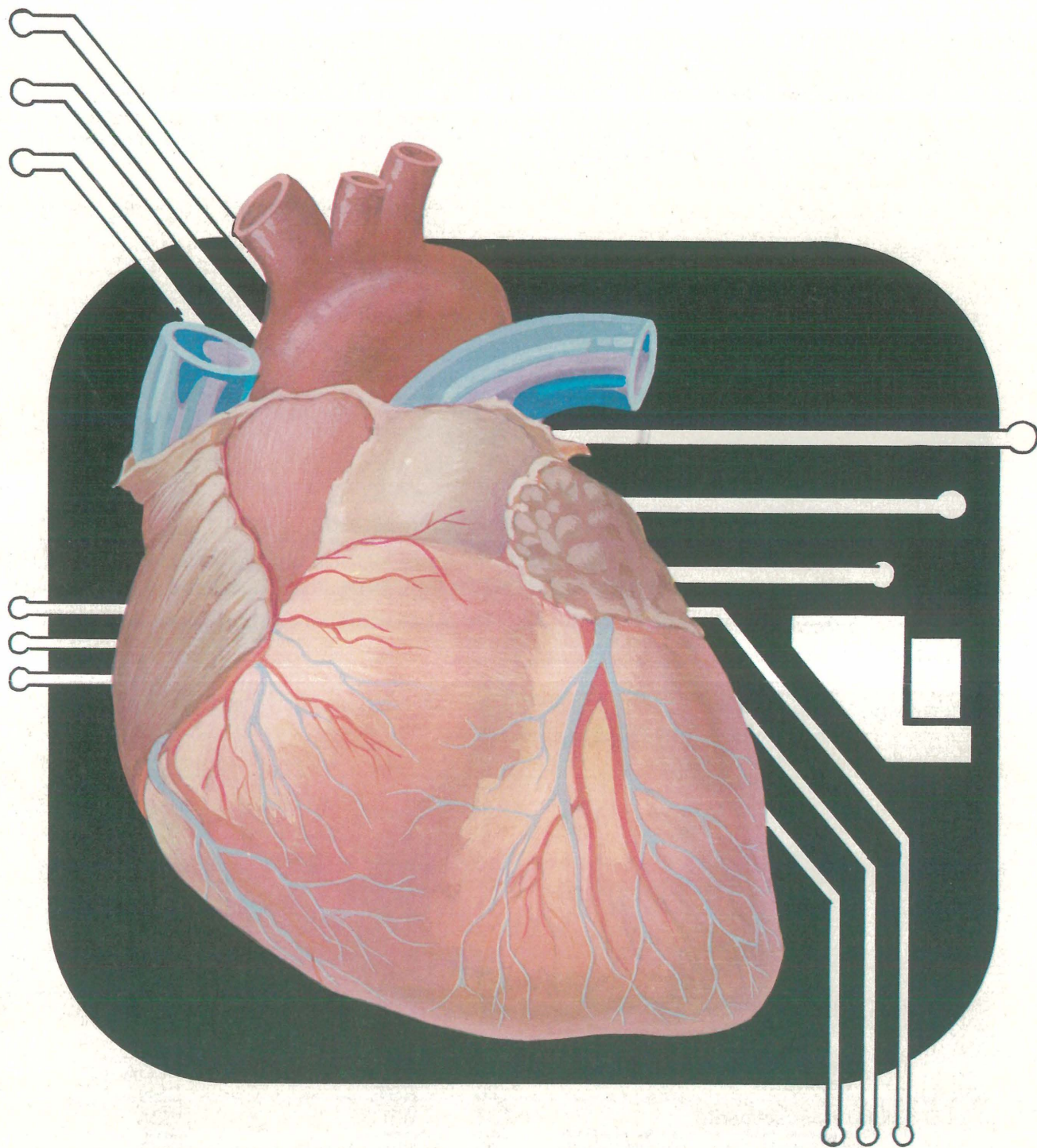
- Segnali di stato, abilitazioni a catena:

INA	(INt. Acknowledge)	riconoscimento di interruzione	(INTACK)
INI	(INt. In)	abilitazione a catena	
INO	(INt. Out)		
HLDA	(HoLD Ackn.)	riconoscimento di HOLD	(HOLDACK)
HLI	(Hold In)	abilitazione a catena	
HLO	(HoLd Out)		
FTC*	(FeTCh)	lettura del primo byte di una istruzione	
ZPE*	(Zero Page Enable)	abilitazione risorse locali della CPU in pagina 0	
RFR	(ReFResh)	ciclo di rinfresco per memorie dinamiche	(REFRESH)
SCK	(System ClocK)	segnale di cadenza (2,4576 MHz)	
LCK	(Line ClocK)	segnale di cadenza (100 Hz)	

* questi segnali sono presenti nella scheda qui descritta, ma non sono definiti nello standard.

- Alimentazioni (per schede 100 × 160, connetto-re 2 × 37):

Connettore MUBUS

Connettore MUBUS per carte formato Europa (100 × 160 mm, conn. 2 × 37 diretto)

Pin nº	Lato componenti	Lato saldature
1	+ 15	+ 15
2	GNDA	GNDA
3	− 15	− 15
4	+ 12	+ 12
5	− 12	− 12
6	LCK	LCK
7	chiave di polarizzazione	
8	0 (ADD)	8 (ADD)
9	1 (ADD)	9 (ADD)
10	2 (ADD)	A (ADD)
11	3 (ADD)	B (ADD)
12	4 (ADD)	C (ADD)
13	5 (ADD)	D (ADD)
14	6 (ADD)	E (ADD)
15	7 (ADD)	F (ADD)
16	$\overline{RFR}$	$\overline{SCK}$
17	$\overline{NMI}$	$\overline{INH}$
18	$\overline{INR}$	$\overline{INA}$
19	$\overline{HLDR}$	HLDA
20	$\overline{RST}$	$\overline{DEN}$
21	$\overline{WR}$	$\overline{VMA}$
22	RDY	$\overline{VPA}$
23	0 (DATA)	$\overline{FTC}$
24	1 (DATA)	ZPE
25	2 (DATA)	DATA 8÷F (per µP 16 bit)
26	3 (DATA)	DATA 8÷F (per µP 16 bit)
27	4 (DATA)	DATA 8÷F (per µP 16 bit)
28	5 (DATA)	DATA 8÷F (per µP 16 bit)
29	6 (DATA)	DATA 8÷F (per µP 16 bit)
30	7 (DATA)	DATA 8÷F (per µP 16 bit)
31	INO (Abilitazione a catena)	INI (Abilitazione a catena)
32	HLO (Abilitazione a catena)	HLI (Abilitazione a catena)
33	− 5	− 5
34	+ 5	+ 5
35	+ 5	+ 5
36	GND	GND
37	GND	GND

Monitoraggio continuo di pazienti gravi in unità coronariche equipaggiate con microcomputer

di **F. Pirri - C. Guerrini** *

Il microcomputer può risultare un valido aiuto nella creazione di strumenti, che rendano più agevole e sicuro il lavoro del medico. In questa nota vengono descritti i principi di funzionamento di un'apparecchiatura per l'analisi di segnali biologici, realizzata dalla General Processor di Firenze.

Introduzione

La strumentazione elettronica tradizionale è di grande aiuto al medico sia nella fase di diagnosi, che nella valutazione quantitativa degli effetti curativi.
Essenzialmente il compito di questi strumenti è la trasformazione in grandezze osservabili direttamente dal medico, di segnali elettrici prelevati direttamente dal corpo del paziente o creati da opportuni trasduttori.
Sui segnali vengono generalmente effettuate solo semplici elaborazioni, quali l'amplificazione ed il filtraggio, demandando al medico ogni funzione interpretativa.
La ragione fondamentale di questo modo di procedere ha ragioni storiche ed è legata principalmente al fatto che anche le più semplici operazioni di riconoscimento, richiedono una considerevole potenza elaborativa, che fino a poco tempo fa era disponibile solo su calcolatori di costo elevato.
Il microprocessore può risultare un valido aiuto nella creazione di strumenti elettronici, che rendano più agevole e sicuro il lavoro del medico.
Esso svolge infatti con facilità compiti tediosi, quali la valutazione delle aree sottese da una porzione di segnale, la misura delle pendenze di una curva, l'accumulo ed il conteggio di dati statistici e così via.
Apparecchiature a microprocessore possono svolgere funzioni di osservazione continua di segnali, estraendo da questi parametri significativi e segnalando eventuali anormalità.
E' bene osservare che nel monitoraggio di pazienti gravi, la funzionalità dello strumento è legata anche al numero di falsi allarmi che vengono generati.
Infatti in genere le risorse infermieristiche sono limitate ed un falso allarme può distogliere il personale da un intervento necessario, inoltre un'apparecchiatura che dia troppi allarmi inesistenti viene a lungo andare trascurata.
D'altra parte, se è relativamente facile rivelare una variazione qualunque del segnale, è invece molto difficile determinare con sufficiente sicurezza quando le variazioni sono dovute a cause non importanti (un picco di rumore, un movimento del paziente etc.) e quando sono invece un sintomo di peggioramento.
E' soprattutto in questo che risulta utile la presenza del microprocessore.
In questa nota vengono sommariamente descritti i principi di funzionamento di un'apparecchiatura per l'analisi di segnali biologici, realizzata dalla General Processor di Firenze.
Lo strumento trae origine da un precedente lavoro svolto nell'Istituto di Elettronica dell'Università degli Studi di Firenze dagli stessi autori.

Acquisizione di segnali biologici

Nel campo dei segnali biologici, il segnale che ha ricevuto la maggiore attenzione dei medici e dei ricercatori è senza dubbio il segnale elettrocardiografico (ECG).
Un normale segnale ECG ha un periodo di ripetizione minore di 1,6 Hz ed anche nelle situazioni patologiche più gravi, come le fibrillazioni cardiache, non si possono superare gli 8 Hz.
Questa situazione è tipica della classe di segnali biologici caratterizzati in genere dal fornire informazioni in tempi lunghi, se confrontati con classi di segnali come quelli usati per esempio nelle comunicazioni elettriche.
Un'altra importante caratteristica del segnale ECG è quella di avere concentrate la maggior parte delle informazioni in brevi intervalli di tempo dell'intero periodo (complessi Q.R.S.).
Questa particolarità può essere ritrovata anche in segnali di origine del tutto diversa: il segnale di un rivelatore di sovra temperatura di un forno presenta delle caratteristiche analoghe, dal punto di vista della significatività delle informazioni fornite.
Questa situazione ha portato ad ideare un sistema a campionamento adattivo, che uscendo dalla rigidità richiesta del Teorema del campionamento di Shannon, potesse effettuare una compressione del numero di campioni necessario per l'analisi di un segnale e possibilmente ne esaltasse il contenuto informativo.
Il sistema da noi creato si basa sulle prestazioni offerte dal microcomputer Child Z.
Al microprocessore infatti facciamo gestire una interfaccia hardware, che lo collega al segnale, e che permette di effettuarne il campionamento, solo quando questo ha rilevanti variazioni di ampiezza.
In altre parole si campiona pochissimo un segnale piatto, mentre si campiona molto più frequentemente la parte di segnale che presenta improvvisi picchi.
Il parametro fondamentale di un tale campionamento adattivo è ovviamente quello che definisce quando una variazione di ampiezza sia rilevante e quando no.
Certamente ogni particolare segnale biologico ha un suo parametro ottimo.
Nei nostri studi ci siamo soffermati in particolare sul segnale ECG.
L'interfaccia realizzata è composta da due converti-

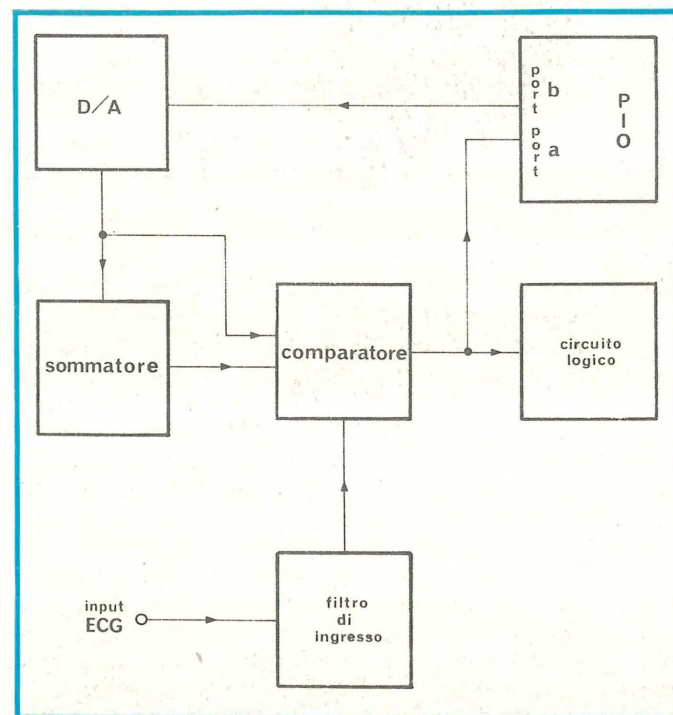



Fig. 1 - Schema a blocchi dell'interfaccia per l'acquisizione di segnali biologici

* F. Pirri - Istituto di Elettronica dell'Università degli studi di Firenze
C. Guerrini - General Processor - Firenze

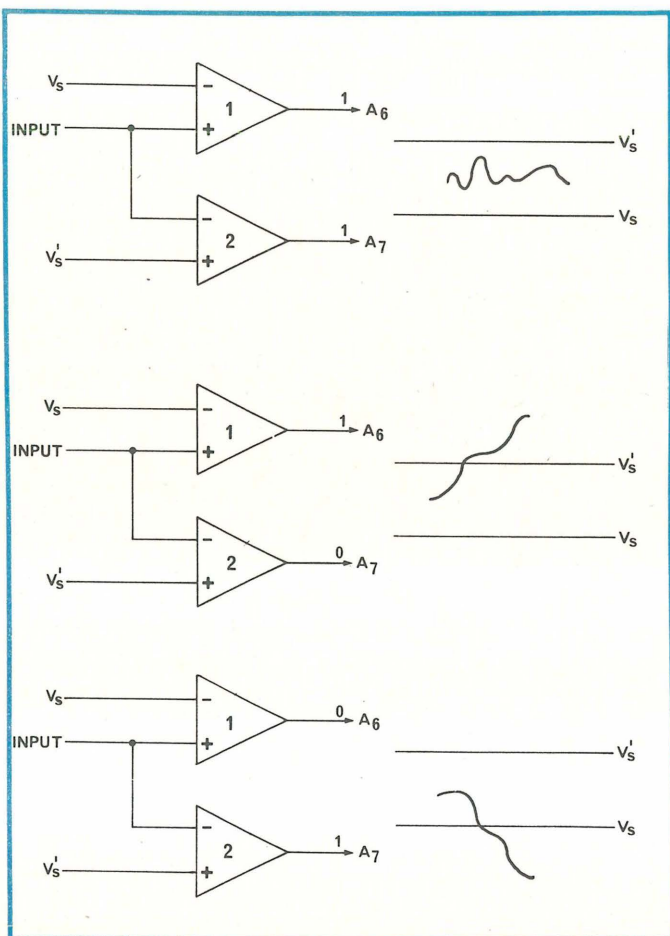



Fig. 2 - Generazione dei segnali di «finestra» per l'inseguimento del segnale

tori D/A, da alcuni amplificatori operazionali e da un comparatore doppio (fig. 1).
Con il comparatore doppio, comandato dal microprocessore attraverso i D/A, si crea una specie di "finestra", che è continuamente spostata, lungo l'asse delle tensioni, con lo scopo di inserirvi in ogni istante il segnale in esame (fig. 2).
Si ha in pratica un continuo esame dell'ampiezza del segnale da parte del microprocessore, che alza od abbassa la finestra, in modo da centrarlo in essa.
La caratteristica di bassa frequenza dei segnali biologici rende trascurabile il tempo impiegato negli spostamenti della finestra, cosicché il segnale viene ad essere composto da una serie di finestre.
In questo modo è possibile sostituire al tratto di segnale compreso tra le due soglie un segmento orizzontale che abbia la stessa durata ed un'ampiezza pari all'ampiezza media della finestra.
La figura 3 mostra come i tratti orizzontali possano ricostruire con sufficiente precisione sia segnali classici come una sinusoide di prova che segnali elettrocardiografici.
In definitiva si ottiene, al posto dei normali campioni equidistanti considerati dal Teorema di Shannon, una piccola serie di coppie AMD (ampiezza media; durata), che permettono di rappresentare il segnale con un numero di campioni inferiore al normale.
Abbiamo in questo modo realizzato una compressione del segnale ed anche una esaltazione del suo contenuto informativo, in quanto si sono eliminati quegli artefatti tipici dei segnali biologici, che danno origine a segnali molto tremolanti e confusi e che sono invece tolti nella sostituzione di un segnale con tanti segmentini orizzontali.

Ricostruzione del segnale ECG

Il passo successivo dei nostri studi è stato l'elaborazione delle n-coppie sequenziali AMD, inviate con il campionamento adattivo al microcomputer.

a

b

c

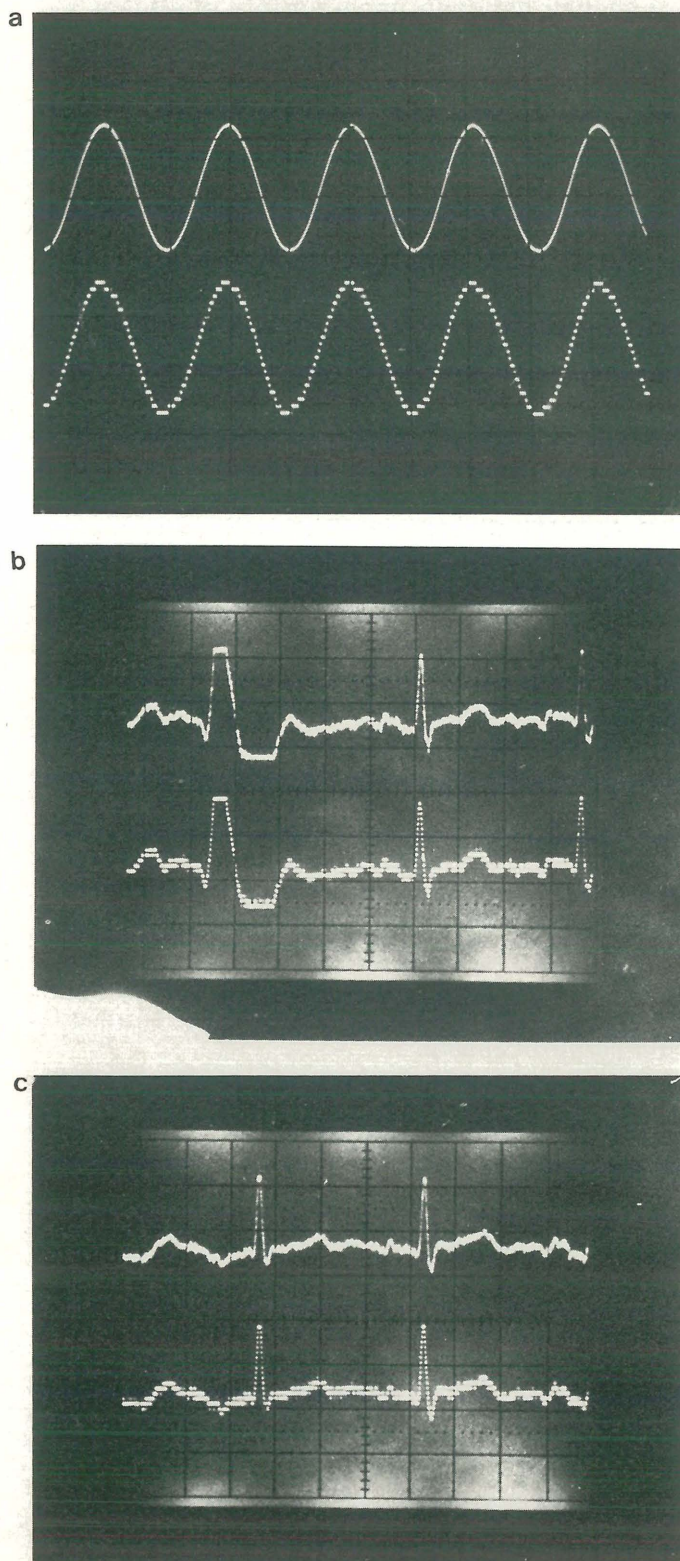



Fig. 3. Ricostruzione di segnali con soli tratti orizzontali:
a) sinusoide
b,c) ECG

Sempre seguendo l'obbiettivo di ottenere il massimo contenuto informativo dal minimo numero di campioni, abbiamo creato un software di ricostruzione per queste n-coppie.

Il programma innanzitutto esamina la durata di ciascun segmentino, ed in base a criteri relativi al tipo di segnale in esame, classifica il segmentino come tratto orizzontale o lo usa per costruirvi delle pendenze. Tutto questo è ottenuto con il confronto della durata con un apposito parametro; segmentini di durata inferiore sono definiti "tratti orizzontali", mentre con gli altri si formano delle pendenze, crescenti o decrescenti, di cui basta memorizzare l'ampiezza in tensione del primo segmentino che le compone, quella dell'ultimo e la durata complessiva di tutti i segmentini. C'è inoltre da dire che in questa fase è inserito un ramo per un'ulteriore eliminazione di artefatti; la presenza infatti di impulsi di rumore di durata molto minore di quelle che sono le normali caratteristiche del segnale, causa infatti la nascita di due tratti orizzontali contigui con uguale ampiezza media; in pratica l'impulso di rumore ha fatto uscire per un attimo il segnale dalla finestra e quando il microprocessore ha operato per reinserirlo in un'altra, ha finito per rimetterlo nella precedente, in quanto il brevissimo impulso di rumore era già terminato.

A ciò si rimedia analizzando continuamente le ampiezze AM di due tratti orizzontali contigui e nel caso risultino uguali, operando una ricomposizione dei tratti artificiali creati dal rumore in un solo effettivo tratto, di durata uguale alla somma delle durate dei precedenti.

Come abbiamo detto, i segmenti che hanno breve durata sono usati per la realizzazione di pendenze; ogni pendenza è costituita da un certo numero di questi segmenti ed ha fine o per l'arrivo di un tratto orizzontale, o per un cambio della sua pendenza.

Si effettua poi un'ulteriore analisi sulla durata dei tratti per classificarli come tratti orizzontali brevi o tratti orizzontali lunghi.

In modo analogo viene studiata l'altezza delle pendenze, per una loro classificazione in lunghe o brevi.

Con questi sistemi si è ricostruito il segnale con tratti orizzontali e pendenze, effettuando un'ulteriore comprensione dei campioni, in quanto come si è detto, più segmentini danno origine ad una sola pendenza e delle ricomposizioni eliminanti il rumore riducono anche il numero dei segmenti che danno origine ai tratti orizzontali.

Con un'accurata scelta dei parametri di classificazione è possibile inoltre ottenere dati interpretabili in modo automatico.

Nel caso ECG, dando al parametro delimitativo delle pendenze un valore che sia superabile solo dai picchi delle onde R, si può avere subito una loro identificazione.

Misurando poi il tempo tra due pendenze lunghe crescenti successive si ha poi anche una misura del periodo cardiaco istantaneo.

E' possibile oltre a ciò effettuare una eliminazione di picchi molto rapidi, che sono causati da artefatti, semplicemente eliminando pendenze lunghe con durata totale troppo breve per poter appartenere al segnale in esame.

Monitoraggio nelle unità coronariche

Il compito principale delle unità coronariche è di tenere continuamente sotto controllo i segnali biologici di un paziente, reduce da un evento grave. Tra i più frequenti di questi c'è l'infarto miocardico; esso può causare nei giorni immediatamente successivi alla sua verifica, complicazioni cardiache gravi e talvolta anche mortali. Uno dei segnali clinicamente più importanti per i pazienti reduci da infarto, è certamente il segnale ECG. Attraverso la sua monitorizzazione continua, si cerca di prevenire le morti causate da eventuali aritmie cardiache. Lown e altri hanno descritto un gruppo di aritmie ventricolari premonitrici, che precedono comunemente la gravissima fibrillazione cardiaca. Quando queste aritmie ventricolari vengono rilevate e curate subito con farmaci antiritmici, l'incidenza di queste nell'arresto miocardico è ridotta al solo 20%. Quando per la rilevazione di aritmie ventricolari, si usano monitors tradizionali, con lettori di intervalli R-R muniti di frequenzimetri a indice o digitali, si va incontro a possibili inconvenienti nel funzionamento. I frequenzimetri ad indice hanno una elevata inerzia e sono sensibili solo a variazioni durature della frequenza cardiaca, non segnalando così nè le sistoli di breve durata (Blocchi senoatriali o pause di 2/3 sec), nè le extrasistole a carattere bigemino. I frequenzimetri digitali invece hanno una eccessiva sensibilità, che fa aumentare l'incidenza dei falsi allarmi, rendendo ancor più pesante il lavoro degli infermieri. A questo punto appare evidente il vantaggio di disporre di un microcomputer, che possa effettuare precise analisi su ogni periodo cardiaco, con particolare riferimento ai complessi ventricolari Q.R.S., e segnalare correttamente eventuali anomalie. Proprio l'incidenza notevole delle aritmie ventricolari può semplificare l'ottenimento di un tale strumento. Si possono infatti realizzare dei programmi che, operando su un segnale ricostruito e curando soprattutto l'analisi degli ampi complessi Q.R.S., diano dei buoni risultati. Per la rilevazione di eventuali anomalie si analizzano in genere diversi parametri. Il principale è il periodo cardiaco istantaneo. Occorre innanzitutto realizzare un sottoprogramma che crei ed aggiorni ad ogni periodo regolare un «periodo medio», con il quale sarà possibile confrontare continuamente quelli in arrivo. Dal confronto potranno scaturire delle valutazioni su eventuali anticipi o ri-

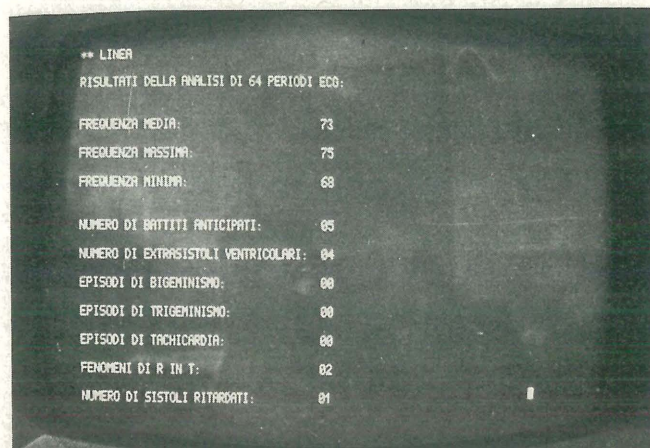



Fig. 4 - Tipica uscita su un terminale video di un programma di analisi ECG al microcomputer

tardi e quindi una individuazione di un gran numero di aritmie ventricolari. C'è da rilevare la necessità dell'aggiornamento continuo del «periodo medio», in quanto altrimenti lente derive della frequenza cardiaca metterebbero fuori taratura il sistema. Altri parametri importanti per la ricerca delle aritmie sono la durata dei complessi Q.R.S., l'ampiezza delle onde R ed eventuali loro cambi di verso. Tutti questi parametri sono ben identificabili con una buona analisi al microcomputer e possono dare risultati pratici molto interessanti.

La figura 4 mostra una tipica uscita generata da un sistema basato sul microprocessore Z80, munito dell'interfaccia sopra descritta ed opportunamente programmato per l'analisi del segnale ECG.

Altre possibili applicazioni

Oltre all'applicazione nel campo delle unità coronariche, molti altri possono essere gli usi di un microcomputer e di un sistema di ricostruzione di segnale come quelli descritti. Trascurando tutti gli altri tipi di segnali, anche solo nel caso dell'ECG, si possono infatti pensare numerose possibilità d'uso. Si può per esempio sfruttare i microcomputer come unità periferiche, per raccogliere dati da passare successivamente ad un minicomputer per le analisi più sofisticate. E' in fase di realizzazione un sistema che permette l'individuazione e la registrazione delle zone di periodi cardiaci nelle quali vi siano delle anormalità, rendendo così possibile una riduzione del lavoro del medico che deve leggersi lunghe registrazioni elettrocardiografiche. Quest'apparecchiatura sarebbe utile anche nella cosiddetta «Elettrocardiografia dinamica», dove attualmente, per avere una riduzione dei dati da passare all'analisi del medico, si effettuano solo brevi registrazioni, a intervalli di tempo fissi, confidando nella statistica per una buona valutazione dei risultati. Indubbiamente l'aggiunta a queste registrazioni casuali, di registrazioni di periodi sicuramente anormali, porterebbe dei miglioramenti nei risultati.

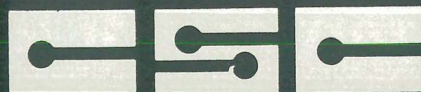
Conclusioni

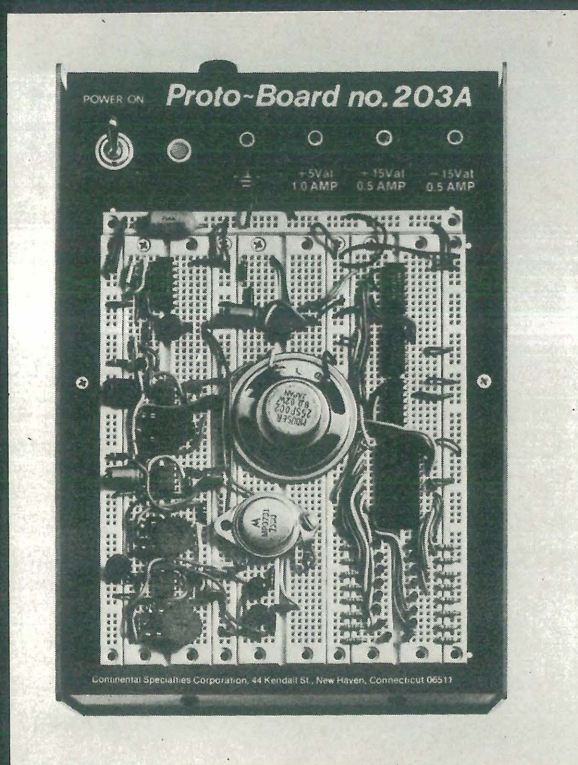
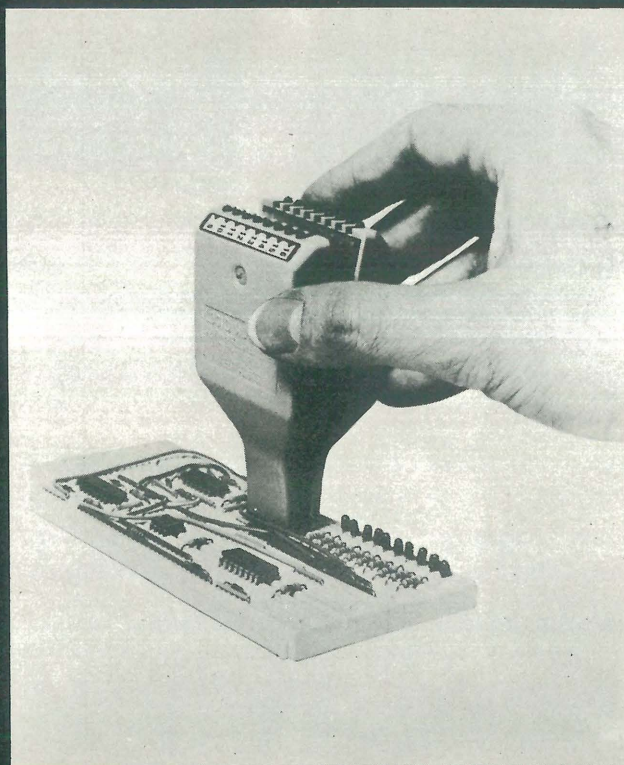
L'elaboratore di segnali biologici sopra descritto trova il suo principale impiego nel monitoraggio continuo nelle unità coronariche. Dopo una sommaria descrizione del principio di funzionamento, si è accennato ad altri possibili usi, quali la preelaborazione per la diagnosi automatica degli ECG e l'elettrocardiografia dinamica. In definitiva, visto il continuo evolversi dei linguaggi e delle capacità dei microprocessori, non è difficile ipotizzare un futuro passaggio ad essi, dei sofisticati programmi per l'analisi automatica completa dell'ECG, e la realizzazione di un'apparecchiatura di tipo ambulatoriale, che sollevi il medico da una parte del lavoro di routine che oggi è obbligato a compiere.

Bibliografia

- LOWN, FAKHRO, HOOD
  « The coronary care unit »
  Jama 199, 1967.
- CACERES, RIKLI
  « Diagnostic computers »
  American lecture series, 717.
- SANTOLI, PELLEGRINI
  « Il trattamento del cardiopatico operato »
  Piccin Editore.
- CORDY, LANG
  « Accuracy of data reduction system for diagnosis and quantification of arrhythmias »
  The American Journal of Cardiology 35, 1975.
- HARRISON, FITZGERALD
  « Ambulatory electrocardiography for diagnosis and treatment of cardiac arrhythmias »
  « Small computer processing of ambulatory electrocardiograms »
  IEEE Procedings, Computer in Cardiology
  Betheseda, 1974.
- RISEMAN
  « P.Q.R.S.T. a guide to electrocardiogram interpretation »
  MacMillan Company. New York.
- CHERRY, BARTON, KUBBA
  « An experimental study of the possible bandwidth compression of visual image signals »
  Proceeding of the IEEE. Novembre 1963.
- OWEN
  « Elettrocardiografia »
  Sigma Tau, 1974.

CONTINENTAL SPECIALTIES CORPORATION

CSC

UN SUCCESSO! COME MAI?

PERCHE' COSTANO MENO

Sistemi di breadboard con o senza alimentatore.

Visualizzatori di stati logici a 16 piedini, TTL/CMOS.

☐ Desidero ricevere il nuovo catalogo C.S.C.

NOME..

COGNOME..

SOCIETÀ..

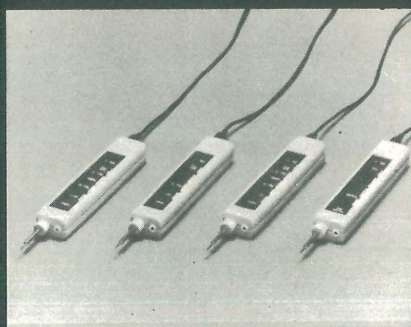
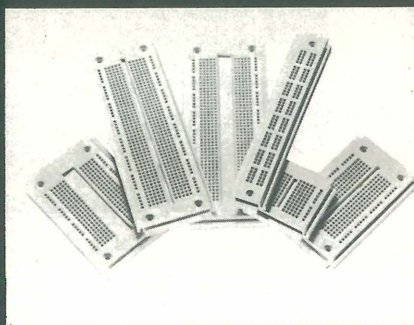
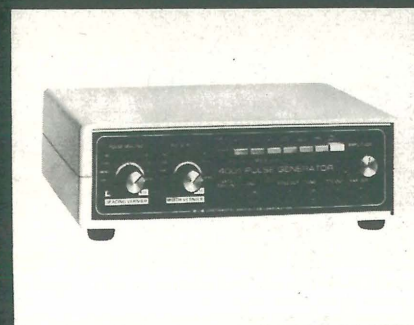
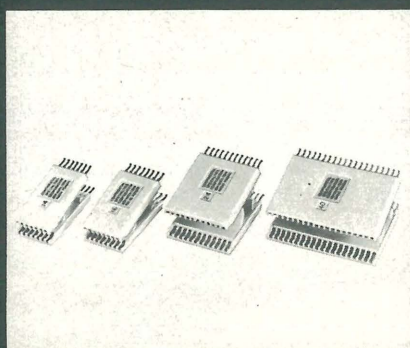
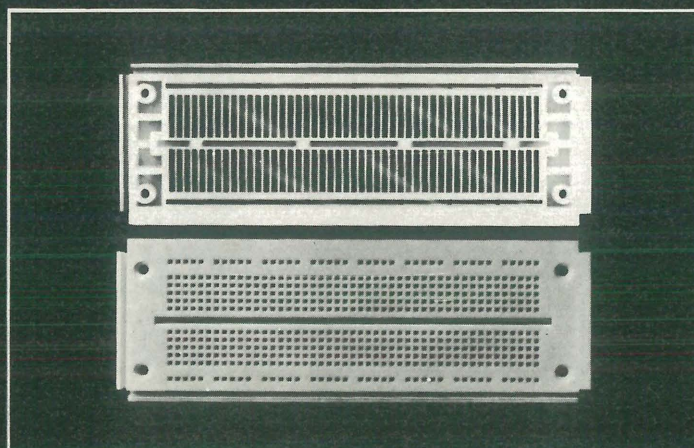
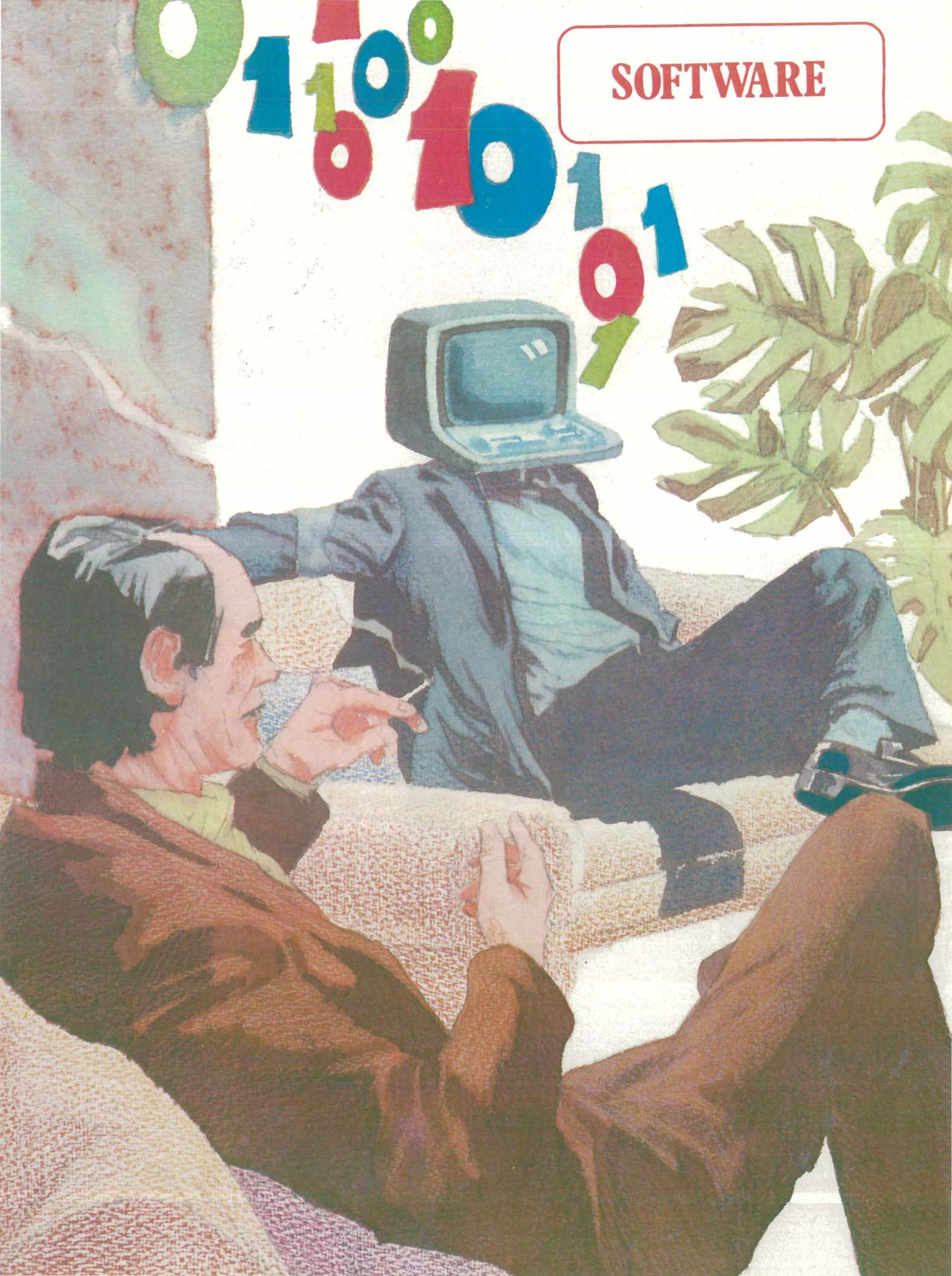
VIA.. TEL. ....

CITTÀ..

CAP.. N. ....

E.O. ....

DISTRIBUTORI

MILANO - FRANCHI TEL. 2894967
TORINO - CARTER TEL. 011 - 592512 597661
GENOVA - ELI TEL. 010 - 581254 565572 565425
FIRENZE - PAOLETTI FERRERO TEL. 055 - 294974 296169
ROMA - SILV ELECTRONICS TEL. 06 - 8313092
SASSUOLO TEL. 059 - 804104 HELLIS
BOLOGNA - ZANIBONI TEL. 045 - 582633
TRIESTE - RADIO KALIKA TEL. 040 - 30341
CATANIA - ELECTRONICS DISTRIBUTION SICILIANA
TEL. 095 682697

SOFTWARE

Text Processing: alcuni cenni

di **T. G. Lewis**

Intrattenersi con un computer come se si trattasse di un essere umano costituisce indubbiamente uno tra gli aspetti più affascinanti delle immense possibilità del computer, anche se per ora si è solo agli inizi.

Introduzione

Un computer può essere programmato per ottenere come risultato delle parole, ossia per generare parole. Supponiamo di voler scrivere un programma che generi in modo casuale N parole di quattro lettere. Scopo del programma è specificatamente quello di riunire le vocali e le consonanti in stringhe, e queste in liste. Il programma cioé comincia a generare una parola scegliendo a caso una vocale o una consonante, concatenando poi la lettera appena scelta alla parola, che è infine memorizzata come stringa. Le funzioni BASIC usate per operare sulle stringhe si presentano in forme diverse a seconda delle diverse versioni BASIC. Alcune tipiche sono:

Concatenazione (+ o &)

L'operatore di concatenazione unisce due stringhe, formando un'unica stringa. Supponiamo che C$ contenga una sola lettera dell'alfabeto, e che W$ sia una parola parzialmente costruita:

W$ = W$ & C$

Questa istruzione di assegnazione (o il suo equivalente in un'altra versione BASIC) produce una parola più lunga, W$, che comprende ora la lettera C$ come lettera finale.

Separazione (SEG o -)

L'inverso della concatenazione è la separazione. Supponiamo di voler estrarre la sottostringa di lunghezza 1, posizione J, dalla stringa A$:

C$ = SEG (A$,J,1)

Questa assegnazione trascrive una singola lettera dalla posizione J di A$ all'interno di C$.

Volendo generare parole di quattro lettere, occorre seguire delle regole: in particolare le regole di scomposizione di una parola sono generalmente parte di una lingua. Ad esempio, nella lingua italiana, la lettera "u" segue la lettera "q" in tutti i casi.

Un modello di regole di scomposizione di una parola può essere costruito osservando come le lettere sono accostate in una lingua per formare parole. Supponiamo di attenerci alle regole seguenti nella generazione di parole di quattro lettere:

1) Ogni parola comincia con una consonante.
2) La seconda lettera è sempre una vocale.
3) La terza e la quarta lettera sono vocali nel 50% dei casi, consonanti nel 50% dei casi.
4) La terza e la quarta lettera sono scelte indipendentemente l'una dall'altra.

Il programma GRAFFITO (v. Fig. 1) genera parole di quattro lettere in conformità alla regole sopraelencate. Esso utilizza le funzioni di stringa descritte prima, e genera il numero richiesto di parole di quattro lettere. Un saggio delle parole generate da GRAFFITO è:

FIBL	WUCC
JIBE	MEOU
TERC	SEET
CUOC	LURE

Chiaramente, le regole assunte in GRAFFITO non sono sufficienti per ottenere parole di quattro lettere

```
100 REM=================================================
110 REM
120 REM          GRAFFITO (PAROLE DA 4 LETTERE)
130 REM
140 REM=================================================
150 DIM A$(5) ,B$(20) ,W$(4) ,C$(1)
160 LET W$=""
170 REM
180 REM --------MEMORIZZA ALFABETO---------------------
190 LET A$="AEIOU"
200 LET B$="BCDFGHJKLMNPRSTVW"
210 REM
220 REM ------SCEGLI IL NUMERO DI PAROLE----------------
230 REM
240 PRINT "QUANTE PAROLE VUOI?";
250 INPUT N
260 FOR N=N TO 1 STEP-1
270 REM
280 REM
290 REM--------RICERCA CASUALE DELLA CONSONANTE----------
300 LET J=INT(17*RND(1)+1)
310 LET C$=SEG(B$,J,1)
320 LET W$=W$&C$
330 REM
340 REM------RICERCA CASUALE DELLA VOCALE----------------
350 LET J=INT(5*RND(1)+1)
360 LET C$=SEG(A$,J,1)
370 LET W$=W$&C$
380 REM
390 REM--------RICERCA DELLA VOCALE/CONSONANTE-----------
400 FOR I=1 TO  2
410 IF RND(1)<0.5 THEN 470
420 LET J=INT(17*RND(1)+1)
430 LET C$=SEG(B$,J,1)
440 LET W$=W$&C$
450 GO TO 500
460 REM----------SCELTA DELLA VOCALE---------------------
470 LET J=INT(5*RND(1)+1)
480 LET C$=SEG(A$,J,1)
490 LET W$=W$&C$
500 NEXT I
510 REM
520 REM
530 PRINT W$
540 LET W$=""
550 NEXT N
560 REM
570 END
```

Figura 1 - Programma GRAFFITO.

Nomi (sintagma soggetto)	Aggettivi	Pronomi	Verbi	Preposizioni	Nomi (oggetti)	Articoli
1	2	3	4	5	6	7
eyes lips caresses love's	hot lewd blue red	your explode	glow radiate	under in on near from	pie sky thigh sigh	the a

Figura 2 - Vocabolario del programma POETRY.

di senso compiuto. Anzi, molto spesso, il risultato è una parola che non dice assolutamente nulla.

Questa esemplificazione ci ricorda che fin dall'inizio dell'era del computer si è cercato di scrivere programmi che capissero la lingua naturale, in quanto si confidava che nel futuro i calcolatori sarebbero stati in grado di capire una lingua. Sfortunatamente, questo scopo non è mai stato pienamente raggiunto, a causa della constatazione che una lingua è più complicata di quanto si possa pensare a prima vista. Ma della strada, sia pure non ben definibile, è stata fatta; scopo dei paragrafi successivi è mostrare alcune immediate anche se molto limitate applicazioni.

Prime considerazioni

Per capire come la lingua opera, bisogna fare riferimento a: 1) *la sintassi*, e 2) *la semantica*. La sintassi si occupa di come le parole sono collegate per formare frasi «corrette». Ad esempio, noi sappiamo per esperienza che *"casa, la rossa, collina sulla grande"* è sintatticamente sbagliato; quello che si vuole intendere è, naturalmente, *"la casa rossa sulla grande collina"*.

Anche se noi di fatto siamo capaci di scoprire il significato di una frase così contorta, l'ordine delle parole trasgredisce la grammatica. La sintassi regola la grammatica del linguaggio, la semantica regola il significato della frase. In entrambi i casi, siamo interessati all'uso non ambiguo di una lingua da parte del calcolatore. Ogni volta che la sintassi è sconvolta, o le parole sono mal scelte, la frase diventa puntualmente più difficile da capire. In questi casi diciamo che la frase è *ambigua*. Si ricordi questa parola, perchè essa ha un ruolo importante nella costruzione di un calcolatore che è in grado di capire una lingua. In certi casi una frase può essere ambigua in un altro senso. Ad esempio, la frase *"gli angoli acuti sul tavolo rotondo ..."* è sintatticamente corretta ma priva di significato. Noi scartiamo senza difficoltà questa frase in quanto espressione inintellegibile, ma una macchina può ravvisare l'assurdità solo per mezzo di ben definite regole di significato.

Frase = sintagma nominale + sintagma verbale
Sintagma nominale = pronome + nome
o pronome + aggettivo + nome
o nome (caso possessivo) + aggettivo + nome
Sintagma verbale = verbo
o verbo + sintagma preposizionale
Sintagma preposizionale = preposizione + articolo + oggetto
o preposizione + articolo + aggettivo + oggetto

Figura 3 - Le regole grammaticali per il programma POETRY.

Di fatto, noi impariamo le regole della grammatica alla scuola elementare; qui ci hanno insegnato delle frasi-tipo; abbiamo imparato che le frasi hanno un *soggetto* e dei *sintagmi verbali*. Il soggetto è costituito normalmente da uno o due nomi, ed eventualmente un aggettivo. I sintagmi verbali sono costituiti da verbi, avverbi, e spesso oggetti diretti e indiretti. In definitiva diciamo che le frasi sono formate in conformità a regole imparate nell'infanzia. Ma come impariamo a parlare?
Per avere una risposta a questo quesito, gli studiosi hanno studiato la grammatica infantile sui bambini di tutto il mondo. Sorprendentemente, hanno scoperto un fatto importante: tutti i bambini, in tutto il mondo, cominciano a parlare usando *la stessa* grammatica! Può darsi che la grammatica di base sia innata, può darsi che la si impari. Non lo sappiamo, ma sta di fatto che tutti i bambini usano lo stesso insieme di regole, indipendentemente dal luogo di nascita, dalla lingua parlata dai genitori, dal colore, dalla religione, dal sesso.

La sola differenza fra lo swahili e l'italiano all'età di 18 mesi è l'insieme di *segni* usati nelle due lingue. I segni sono le parole reali (parlate o scritte) usate in senso espressivo. In swahili, potremo dire *"me-me"* anzichè *"io"*, ma la grammatica è la stessa nelle due lingue *fino ai 18 mesi di età.*

Sintassi infantile

Una raccolta di segni (parole) forma un *vocabolario* che chiameremo V_T:

$$V_T = (scarpa,\ caldo,\ latte).$$

In questo esempio, V_T è costituito da singole parole. Queste parole sono messe insieme in italiano (o i loro equivalenti in francese, inglese, tedesco, etc.) me-

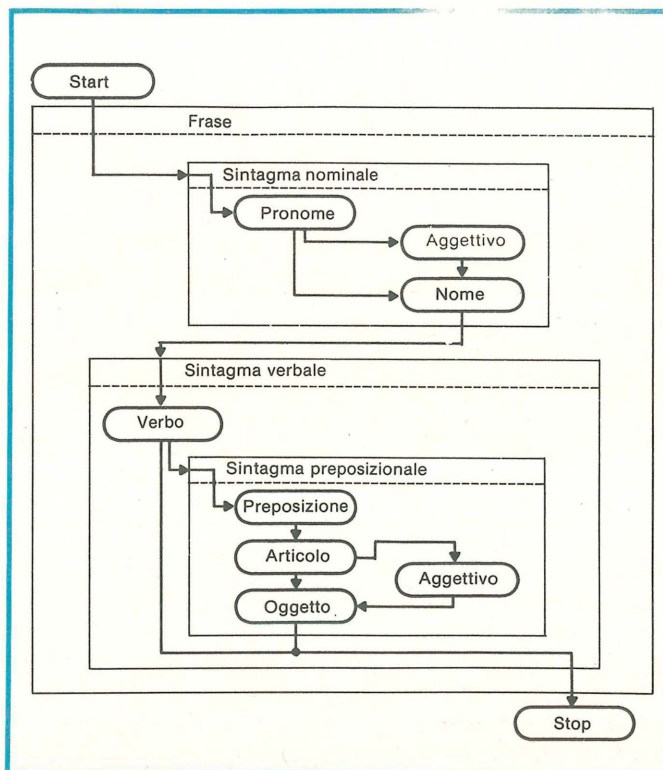



Figura 4 - Diagramma della grammatica del programma POETRY.

(A) Programma principale

```
10 PAGE
100 REM ==================================
110 REM
120 REM          POETRY
130 REM
140 REM----------------------------------
150 LET M=240
160 LET K=1
170 LET S=1
180 LET L=8
190 DIM V$(M),V(8),W$(8)
200 LET V(1)=1
201 LET V$=""
205 REM-------------------------------
210 PRINT "ENTER NOUNS(8 CHAR)"
220 GOSUB 1000
230 PRINT "ENTER ADJECTIVES(8 CHAR)"
240 GOSUB 1000
250 PRINT "ENTER PRONOUNS(8 CHAR)"
260 GOSUB 1000
270 PRINT "ENTER VERBS (8 CHARACTERS)"
280 GOSUB 1000
290 PRINT "ENTER PREPOSITIONS (8 CHAR)"
300 GOSUB 1000
310 PRINT "ENTER OBJECTS (8 CHAR)"
320 GOSUB 1000
330 PRINT "ENTER ARTICLES(8 CHAR)"
340 GOSUB 1000
350 REM --------GENERATE SENTENCES-------
360 GOSUB 2000
370 END
380 REM
390 REM
400 REM
410 REM
```

(B) Sottoprogramma di Input

```
1000 REM===============================
1010 REM
1020 REM        INPUT VOCABULARY
1030 REM
1040 REM-------------------------------
1045 LET S=S+1
1050 REM---REPEAT LOOP ---------------
1060 PRINT "WORD=";
1070 INPUT W$
1075 IF W$="" THEN 1230
1080 REM---PACK WORD INTO VOCABULARY--
1110 LET V$=V$&W$
1111 FOR I=LEN(W$)+1 TO L
1112 LET V$=V$&" "
1113 NEXT I
1120 REM---MARK BEGINNING OF UNIT ----
1130 LET V(S)=K+L
1140 LET K=K+L
1150 IF K<M THEN 1050
1160 PRINT "VOCABULARY FULL"
1230 RETURN
```

(C) Generatore di frase

```
2000 REM===============================
2010 REM
2020 REM       GENERATE SENTENCES
2030 REM
2040 REM-------------------------------
2050 PRINT "ENTER NUMBER OF SENTENCES YOU WANT=";
2060 INPUT N
2070 FOR N=N TO 1 STEP -1
2080 REM---NOUN PHRASE----------------
2090 GOSUB 3000
2100 REM---VERB PHRASE----------------
2110 GOSUB 4000
2120 PRINT
2130 NEXT N
2140 RETURN
2150 REM------------------------------
```

(D) Generatore di sintagma nominale

```
3000 REM==============================
3010 REM
3020 REM     NOUN PHRASE
3030 REM
3040 REM------------------------------
3050 LET S=3
3060 REM---GET PRONOUN----------------
3070 GOSUB 5000
3080 IF RND(1)<0.2 THEN 3500
3090 LET S=2
3100 REM---GET ADJECTIVE--------------
3110 GOSUB 5000
3120 REM---GET NOUN-------------------
3500 LET S=1
3510 GOSUB 5000
3520 RETURN
```

(E) Generatore di sintagma verbale

```
4000 REM==============================
4010 REM
4020 REM     VERB PHRASE
4030 REM
4040 REM------------------------------
4050 LET S=4
4055 REM---GET VERB-------------------
4060 GOSUB 5000
4070 IF RND(1)<0.1 THEN 4500
4080 LET S=5
4090 REM---GET PREPOSITION------------
4100 GOSUB 5000
4110 LET S=7
4120 REM---GET ARTICLE----------------
4130 GOSUB 5000
4140 IF RND(1)<0.3 THEN 4400
4150 LET S=2
4160 REM---GET ADJECTIVE--------------
4170 GOSUB 5000
4400 REM---GET OBJECT-----------------
4410 LET S=6
4420 GOSUB 5000
4500 RETURN
```

(F) Output di una parola per volta

```
5000 REM==============================
5010 REM
5020 REM      PRINT THE WORD
5030 REM
5040 REM------------------------------
5050 LET N1=(V(S+1)-V(S))/L
5060 LET N1=INT(RND(1)*N1)
5070 LET W$=SEG(V$,V(S)+N1*L,L)
5080 PRINT W$;
5090 RETURN
```

Figura 5 - Programma POETRY.

diante la grammatica delle singole parole da tutti i bambini fino ai 18 mesi. Esprimiamo la regola in questo modo:

Frase = parola, dove la parola è tratta da V_T.

Diciamo che la frase di un bambino è costruita scegliendo una parola dal vocabolario V_T.
Chiaramente, la lingua sarebbe una ricerca priva di interesse se consistesse solo di frasi di una sola parola. All'età di 18 mesi circa, la maggior parte dei bambini sviluppa una grammatica di due parole, consistente in due specie di parole. Le *parole aperte* sono segni che vengono prelevati dall'elenco delle singole parole:

Parole aperte = (*scarpa, caldo, latte*).

Le *parole cardine* vengono aggiunte a V_T ; ad esempio,

Parole cardine = (*mio, prato*)

La grammatica vuole che un bambino scelga parole da un V_T fatto di parole aperte e parole cardine, come segue:

Frase = parola aperta + parola aperta

oppure

Frase = parola cardine + parola aperta

In altri termini la struttura della frase (sintassi) di un bambino di 18 mesi, in Africa, in Cina o in Europa, è costituita da due parole entrambe aperte o da una parola cardine seguita da una parola aperta. Non si danno altre combinazioni.

Considerando il caso di un bambino che dispone del vocabolario di V_T = ((parole aperte), (parole cardine)), si possono dare i seguenti esempi di espressioni:

Frase = parola aperta + parola aperta
= *"latte" + "caldo"*
= *"latte caldo"*

LOVE'S	BLUE	CARESSES	GLOW	FROM	A	LEWD	THIGH
YOUR	HOT	LIPS	GLOW	ON	A	LEWD	SIGH
YOUR	LEWD	EYES	EXPLODE	FROM	THE	BLUE	THIGH
YOUR	HOT	CARESSES	EXPLODE	UNDER	THE	RED	SIGH
YOUR	RED	CARESSES	GLOW	ON	A	RED	SKY
YOUR	BLUE	LIPS	RADIATE	UNDER	THE	RED	THIGH
YOUR	EYES	RADIATE	ON	THE	BLUE	THIGH	
YOUR	BLUE	EYES	RADIATE	ON	A	SKY	
YOUR	LIPS	EXPLODE	NEAR	THE	SIGH		
LOVE'S	RED	CARESSES	GLOW	FROM	THE	RED	SKY
YOUR	HOT	EYES	RADIATE				
LOVE'S	RED	CARESSES	RADIATE	IN	A	HOT	PIE

Le azzurre carezze d'amore ardono da una impudica coscia
Le tue calde labbra ardono su un impudico sospiro
I tuoi impudichi occhi scoppiano dalla azzurra coscia
Le tue calde carezze scoppiano sotto il rosso sospiro
Le tue rosse carezze ardono su un rosso cielo
Le tue azzurre labbra risplendono sotto la rossa coscia
I tuoi occhi risplendono sulla azzurra coscia
I tuoi azzurri occhi risplendono su un cielo
Le tue labbra scoppiano presso il sospiro
Le rosse carezze d'amore ardono dal rosso cielo
I tuoi caldi occhi risplendono
Le rosse carezze d'amore risplendono in un caldo disordine

Figura 6 - Frasi generate dal programma POETRY. La traduzione è letterale.

Un'altra frase che il bambino può formulare è derivata nello stesso modo, ma con parole diverse:

Frase = parola cardine + parola aperta
= *"mio" + "latte"*
= *"mio latte"*

Poichè il bambino non può mettere insieme in una sola frase due parole cardine, non potrebbe mai dire *"prato mio"*. Questo si spiega con le regole seguenti:

"prato mio" = parola cardine + parola cardine

Frase ≠ parola cardine + parola cardine

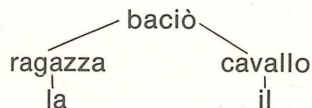
Quando sono ormai padroni di questa semplice grammatica, i bambini, in tutto il mondo, progrediscono rapidamente verso una grammatica che ammette quattro specie di frasi:

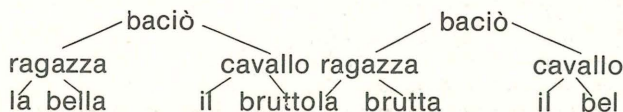
Frase = parola aperta
o parola aperta + parola aperta
o parola aperta + parola cardine
o parola cardine + parola aperta

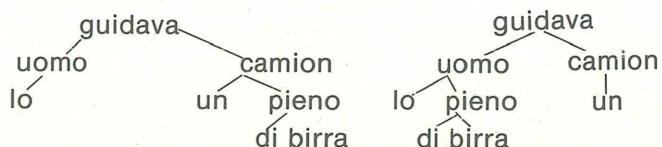
Attorno ai 3 anni, l'influenza della cultura comincia ad agire sulla capacità innata del bambino di formare frasi.
Quando il bambino comincia ad usare tre parole in una frase, impara a formarle secondo le regole della sua lingua natale. È infatti in questa fase dello sviluppo del bambino che cominciano a delinearsi le differenze tra le varie lingue.

Analisi sintattica di una frase

In un certo senso, possiamo vedere gli esseri umani come operatori sintattici. Avendo continuamente a che fare con delle frasi, modifichiamo, analizziamo, correggiamo, e in qualche modo facciamo nostra la struttura sintattica e la investiamo di significato. La frase *"la ragazza baciò il cavallo"* genera immediatamente invidia per il cavallo quando analizziamo la frase e ci rendiamo conto (attraverso un'immagine mentale, probabilmente) di quello che essa vuol dire. Quindi *analizziamo* o "identifichiamo" la frase sia in senso sintattico che in senso semantico.
Volendo programmare un calcolatore per analizzare una frase, dovremo costruire un diagramma della frase, detto *indicatore sintagmatico*. Un indicatore sintagmatico per la frase "la ragazza baciò il cavallo" è il seguente:

baciò
ragazza cavallo
la il

Se identifichiamo (analizziamo) la frase modificata "la bella ragazza baciò il brutto cavallo", o "la brutta ragazza baciò il bel cavallo", avremo due differenti indicatori sintagmatici, come si può vedere qui sotto:

baciò baciò
ragazza cavallo ragazza cavallo
la bella il brutto la brutta il bel

Negli indicatori sintagmatici riportati sopra non abbiamo problemi per decidere quale dei due attori è bello e quale è brutto. Gli indicatori sintagmatici sono chiaramente unici, e diciamo che sono *non ambigui*.

Consideriamo ora la frase ambigua *"L'uomo guidava un camion pieno di birra"*. I due possibili indicatori sintagmatici sono i seguenti:

Poiché non possiamo determinare quale dei due indicatori sintagmatici è "corretto", la frase non può essere analizzata ricorrendo ad un unico indicatore sintagmatico. Di conseguenza, è una frase ambigua. Come possiamo aspettarci che un calcolatore capisca le frasi ambigue, se non le capiamo neppure noi? Ovviamente, per chiarire il significato di una frase, noi siamo abituati a fare delle ipotesi, a ricorrere al contesto, e ad operare degli aggiustamenti in base a tutta una serie di riferimenti culturali noti. Sappiamo ad esempio che *"Giorgio ha fatto le scarpe al suo collega"* vuol dire che ha fatto in modo di danneggiarlo, non che gli ha confezionato un paio di scarpe. Perciò dobbiamo fornire al calcolatore le regole della semantica perché possa capire la lingua come noi.

Un esempio

Per poter programmare un calcolatore perché capisca una lingua, dobbiamo conoscere le regole della grammatica e le regole della semantica. Le regole della grammatica sono formulate con chiarezza, quelle della semantica invece sono più difficili da individuare. Ci fermeremo quindi all'aspetto più semplice del problema, cioè alle regole grammaticali necessarie per formare le frasi. Per produrre meccanicamente frasi grammaticalmente corrette, abbiamo bisogno di un modello di grammatica. Ricordiamo che la grammatica regola la collocazione delle parole nelle frasi; dunque per cominciare abbiamo bisogno sia di una grammatica che di un vocabolario.
Spieghiamo concretamente: il vocabolario di Figura 2 è immesso in un programma che scrive componimenti poetici in inglese. Ciascuna riga di un componimento poetico può essere costituita da nomi, aggettivi, pronomi, verbi, preposizioni ed articoli. Come si può vedere, si tratta di poesia del "genere romantico"; notiamo inoltre che è presente una serie di oggetti rimanti fra loro. Possiamo programmare un calcolatore perché produca poesia "originale" sulla base di questo limit. o vocabolario, usando come modello grammaticale l'indicatore sintagmatico. Usiamo questo modello per generare frasi che suonino "corrette" nella logica della poesia romantica. Il modello è espresso in una notazione esattamente identica a quella impiegata per descrivere il linguaggio dei bambini. La serie delle regole grammaticali in questo caso è detta sistema di *produzione generativa*.
Le quattro semplici regole di Fig. 3 sono tutto quello di cui abbiamo bisogno per produrre una poesia "intellegibile". Un altro modo di vedere questo modello si ricava dal diagramma di Figura 4. Il diagramma è utile nella scrittura del programma del calcolatore, perché mostra come generare ciascuna parola in ciascuna riga della poesia. Cominciando da START, seguiamo la freccia fino a SINTAGMA NOMINALE. Il SINTAGMA NOMINALE conduce a PRONOME, che a sua volta conduce a NOME nel 20% delle frasi, ad AGGETTIVO nell'80% delle frasi. Ogni AGGETTIVO è seguito da un NOME ed ogni NOME è seguito da un SINTAGMA VERBALE. Così possiamo generare righe di poesia, ognuna diversa dalle altre, seguendo le frecce del modello di Figura 4.

I programmi in BASIC di Figura 5 percorrono esattamente il cammino che abbiamo illustrato. Abbiamo fatto girare questo programma su un calcolatore dotato di un generatore di numeri casuali per produrre una selezione casuale di parole dal vocabolario e una selezione casuale di percorso.
Ciascun componimento poetico è costituito da un numero di frasi indicato da un valore d'ingresso. Le parole sono lunghe 8 caratteri (con spazi vuoti intercalati) e sono memorizzate come stringhe in una lista, V$. Il vettore V di 8 elementi contiene gl'indici di V$ corrispondenti all'inizio dell'elenco di nomi, aggettivi, etc., contenuti in V$. Quindi S è il valore dell'indice dell'inizio di ciascun elenco.
Poiché abbiamo presupposto un calcolatore capace di elaborare parole semplici, il programma di Figura 5 è usato per generare le parole una alla volta.
Un campione di 12 frasi di questa poesia sui generis è riportato in Figura 6.
Il modello generativo presentato non affronta il problema dell'ambiguità, perché è originato da un calcolatore, non da un essere umano. Il calcolatore obbedisce a regole rigide specificate nel programma. Per costruire un calcolatore che "capisca" la poesia che esso genera, sarebbe necessario un programma molto più sofisticato. ■

PERSONAL COMPUTER PET 2001
MICROCOMPUTER CMB
SERIE 3001

Sistemi completi con unità centrali da 8, 16, 24, 32K RAM-Video-Memoria a cassette magnetiche e floppy-disk. Stampanti da 40-80-132 colonne. Interfaccie varie.

Distribuzione HARDEN S.p.A. - Div. Elettr. - 26048 SOSPIRO (Cremona) Tel. (0372) 63136 r.a. - TLX 320588

Interprete Basic in 8080

I - Parte di **F. Maddaleno**

Introduzione

Il programma qui presentato è un interpretatore Basic ridotto chiamato Tiny Basic Interpreter (TBI); è stato tratto dalla rivista 'Dr. Dobb's Journal of Computer Calisthenics & Orthodontia' del maggio '76, ed è stato modificato per renderlo rilocabile. Il TBI occupa 2 kbyte di memoria (RAM o ROM), e richiede della RAM per il programma di utente. La quantità della RAM richiesta può essere facilmente modificata per poter adattare il programma al sistema disponibile. Il TBI è scritto in codice 8080 e può quindi funzionare anche su Z80: avendo a disposizione quest'ultimo microprocessore e relativo assemblatore, il TBI può ancora essere accorciato per ottenere più spazio per eventuali aggiunte: si è voluto mantenere il codice 8080 poichè questo microprocessore è molto diffuso.
Il TBI è stato scritto in modo da ottenere la massima compattezza di codice, mentre non si è dato peso alla leggibilità del programma: esso è quindi una buona collezione di 'trucchi' di programmazione ed un ancor migliore esempio di programmazione *non* strutturata.
Si vuole ricordare qui di passaggio la differenza fra interpretatore e compilatore: quest'ultimo è un programma che accetta in ingresso un sorgente in linguaggio evoluto (es. Fortran) e lo traduce in assembler, poi si assembla la traduzione e si fa eseguire l'oggetto così prodotto. Il lavoro di traduzione ed assemblaggio è fatto prima dell'esecuzione, ed una volta sola: quando si esegue il programma, il compilatore non deve essere presente in memoria: addirittura si può eseguire la compilazione su una macchina diversa da quella su cui girerà il programma. L'interpretatore è invece un programma che deve essere presente in memoria al tempo di esecuzione insieme al programma di utente memorizzato ad esempio in caratteri ASCII: l'interpretatore legge una linea di programma di utente, riconosce in essa dei comandi, li esegue appena riconosciuti e passa alla linea seguente: in un loop ad esempio deve quindi 'capire' ogni volta cosa significa una istruzione e poi eseguirla: in fase di esecuzione è quindi più lento un programma interpretato di uno compilato, ma si ha il vantaggio che un interpretatore è più semplice e più corto di un compilatore.

GAME TORINO

a cura del Gruppo Amatoriale Micro-Elaboratori, Torino

Linguaggio accettato dal TBI

L'aritmetica del TBI è intera ed i numeri devono essere compresi fra −32767 e +32767. Ci sono 26 variabili chiamate con le lettere maiuscole da A a Z, ed un vettore@(indice) che occupa tutta la RAM lasciata libera dal programma d'utente. Il TBI possiede tre funzioni matematiche:

- ABS (<expr>) che dà il valore assoluto dell'espressione.
- RND (<expr>) che fornisce un numero casuale uniformemente distribuito fra 1 e il valore di <expr>, che deve essere maggiore di zero.
- SIZE che dice il numero di byte lasciati liberi dal programma d'utente: quindi considerando che ogni numero occupa due byte e che la RAM libera può venire occupata dal vettore @ si vede che il valore SIZE/2 dice la dimensione massima del vettore con quel programma in memoria.

Nel TBI ci sono i seguenti operatori aritmetici e di relazione:

/	per indicare la divisione
*	per indicare la moltiplicazione
−	per indicare la sottrazione
+	per indicare l'addizione
− unario	per indicare il segno negativo
+ unario	per indicare il segno positivo
=	per indicare uguale
>=	per indicare maggiore o uguale
>	per indicare maggiore
<=	per indicare minore o uguale
<	per indicare minore
#	per indicare non uguale

Il risultato di un confronto è uno oppure zero a seconda che la relazione sia vera o falsa.
Espressioni
Le espressioni sono formate da numeri, variabili e funzioni collegati da operatori aritmetici e logici. Le espressioni sono valutate da sinistra a destra, dando però la precedenza a moltiplicazioni e divisioni, poi ad addizioni e sottrazioni ed infine agli operatori di relazione: questi ultimi possono essere usati in qualunque espressione.
Per variare l'ordine di valutazione delle espressioni si possono usare le parentesi con le regole solite dell'aritmetica: il numero massimo di parentesi che possono essere annidate dipende solo dalle dimensioni dello stack. La grammatica delle espressioni riconosciute dal TBI è riportata nel listato.

Esempi: 20 LET A= (X>Y)*X + (X= Y)*(−33) + (X<Y)*Y

```
40 IF (U= 1) *(V<2)+(U>V)*(U<99)*(V>3) PRINT
   "QUESTO"
60 LET R=RND(100, A= (R>3) + (R>15) + (R>56)
   + (R+98)
```

Nell'istruzione 20 ad A viene assegnato il valore del maggiore fra X e Y, e se sono uguali ad A viene dato il valore −33.
Nell'istruzione 40 il simbolo di moltiplicazione funziona come and logico e quello di addizione come or logico (vedere anche il comando IF).
Nell'istruzione 60 la variabile A è un numero casuale con probabilità del 3% di valere zero (nessuna condizione verificata, cioè R non maggiore di 3), del 12% di valere 1 (R compreso fra 4 e 15), del 41% di valere 2, del 42% di valere 3 e del 2% di valere 4: anche qui si sfrutta il fatto che il risultato di un confronto può essere zero o uno e questo numero può entrare in qualunque punto di una espressione.

Comandi

L'interpretatore accetta due tipi di comandi: diretti e non diretti. I comandi diretti sono quelli che vengono interpretati ed eseguiti appena sono stati battuti, senza essere messi in memoria, gli altri invece vengono messi in memoria in attesa di essere interpretati al tempo di esecuzione. Tutti i comandi qui descritti possono essere usati come comandi diretti o messi in un programma, eccetto i primi tre che possono solo essere usati come comandi diretti.

RUN

Fa partire l'esecuzione del programma messo precedentemente in memoria a partire dall'istruzione con numero più basso.

LIST e LIST< num>

Il primo fa stampare tutto il programma, con le istruzioni ordinate secondo il loro numero crescente, il secondo fa stampare il programma a partire dall'istruzione con numero uguale o maggiore di quello dato.

NEW

Cancella tutto il programma presente in memoria.
I comandi precedenti possono solo essere diretti, mentre quelli successivi possono esserlo o no. Il TBI interpreta uno dei comandi successivi come diretto quando non si trova in una istruzione, e quindi lo esegue subito. Quando invece il comando si trova in una istruzione, viene salvato in memoria. Le istruzioni sono linee formate da un numero compreso fra 1 e 32767 seguito da uno o più comandi separati da punto e virgole ';', la linea è poi chiusa da un ritorno carrello (CR). I comandi GOTO, STOP, RETURN devono essere gli ultimi in una istruzione. Tutte le linee, sia per le istruzioni che per i comandi diretti, devono essere più corte di 72 caratteri e devono essere chiuse da un CR.

Negli esempi che seguono si è sempre tralasciato il numero della linea e l'eventuale ';' finale che separa il comando da un altro sulla stessa linea.

REM

Serve per introdurre i commenti: qualunque cosa segua la REM viene ignorata dal TBI; l'interpretazione riprende alla riga successiva.

Es. REM COMMENTO QUALUNQUE

LET

Le assegnazioni possono avere i seguenti formati

<let> :: = LET <assegnazioni> / <assegnazioni>
<assegnazioni> :: = <assegnazione> / <assegnazioni>, <assegnazione>
<assegnazione> :: = <var> = <expr> / <vett> = <expr>

Il resto della grammatica si trova nel listato alla routine EXPR.

Esempi LET A = 234-5*6, A = A/2, X = A-100, @(X + 9) = A-1

Questa istruzione assegna ad A il valore di 234-5*6 cioè 204, assegna di nuovo ad A il valore di A/2 cioè 102, assegna alla variabile X il valore di A-100 cioè 2 ed infine pone l'undicesimo elemento del vettore a 101, poiché l'indice del vettore vale X + 9 cioé 11 e A-1 fa 101.

LET U=A ≠ B, V=(A>B)*X+(A<B)*Y

Pone la variabile U a 1 se A è diverso da B e a zero se A = B, poi assegna a V il valore di X o Y o zero a seconda che A sia maggiore, minore o uguale a B.
A = 5
B = A + C, D = G + H
Queste istruzioni assegnano ad A il valore 5, a B e D il valore di A più C e G più H rispettivamente: notare che non è presente il LET.

PRINT
Il comando PRINT serve per controllare l'output del programma.
Può essere seguito o no da una lista di voci. Quando non è seguito da una lista di voci, fa uscire un CR ed un line-feed (cioè va a capo). Le voci che possono seguire il PRINT sono espressioni, stringe fra apici, stringhe fra virgolette, formati e back-arrow, che devono essere separati fra di loro da virgole. Il comando PRINT stampa il valore delle espressioni e le stringhe tenendo conto dei formati e dei back-arrow.
Esempi PRINT A/3 + 1, "AB'CD", 'VG"BH'
Stampa il valore dell'espressione, la stringa AB'CD, e la stringa VG"BH: si nota che si può usare come delimitatore di stringa sia 'sia", però la stringa deve essere chiusa con lo stesso delimitatore con cui è stata aperta.
PRINT A/3 + 1, "AB'CD", 'VG"BH',
Stampa le stesse cose del precedente comando ma alla fine non va a capo (c'è la virgola finale): si può così continuare a stampare sulla stessa riga con un altro comando di PRINT.
PRINT A, B, ≠ 3, C, D, E, ≠ 10, F, G, − 567
Stampa il valore di A e di B riservando loro 6 spazi (valore che il TBI assume automaticamente se non diversamente indicato), stampa poi i valori di C D ed E usando 3 spazi per ciascuno, infine stampa in 10 spazi per uno i valori di F G e il numero − 567. Se il formato è troppo piccolo per il valore da stampare

(es. C vale 2222 oppure −456) il valore viene comunque stampato tutto. Se il numero di spazi è maggiore di quello delle cifre da stampare, vengono aggiunti dei blank a destra. Dopo il simbolo di formato ≠ ci può essere una espressione: del valore viene solo preso il byte meno significativo.

PRINT 'ABC', ← , 'XXX'

Stampa la stringa ABC poi fa un ritorno carrello senza passare alla linea successiva e stampa sopra alla stringa ABC la nuova stringa XXX poi va a capo normalmente.

INPUT

Serve per introdurre dati dall'esterno durante l'esecuzione.

Esempi   INPUT A, B

Questo comando fa uscire sulla stampante A: poi il TBI si mette in attesa di una espressione battuta da tastiera, il cui valore è assegnato ad A. Poi viene stampato B: e si ripete la sequenza. Se l'espressione battuta contiene degli errori il TBI segnala l'errore e ripropone la domanda fino a che non viene battuta una espressione corretta.

Dopo l'INPUT ci possono essere stringhe fra apici o virgolette o back-arrow (separati da virgole) che si comportano esattamente come nella PRINT.

INPUT 'DATA', X, 'ORA'Y

Fa stampare DATAX: e ORA: Nel secondo caso non compare il nome della variabile perché non c'è la virgola fra la variabile e la stringa che la precede.

IF

Il comando IF può avere i seguenti formati:

<if> :: = IF <expr> <comandi>

<comandi> :: = <comando> / <comandi>; <comando>

Se l'espressione ha valore zero, il TBI salta alla linea di programma successiva. Se l'espressione è invece diversa da zero (e non solo quando vale uno) vengono eseguiti i comandi (o il comando) presenti sulla linea dopo l'espressione: se vi sono più comandi, questi devono essere separati fra di loro da ';'. Fra questi comandi possono esservi degli altri IF. Il TBI passa poi alla linea seguente.

Esempio IF (A>B) + (C=D) R=55; PRINT 'QUESTO MESSAGGIO'

Se A è maggiore di B oppure C è uguale a D, R viene posto a 55 e viene stampata la stringa, poi il controllo passa alla linea dopo; se invece il valore dell'espressione è zero, il controllo passa immediatamente alla linea successiva.

GOTO

Il comando di salto é GOTO < expr > dove l'espressione indica il numero della linea a cui deve passare il controllo. Il comando GOTO (e non GO TO) non può essere seguito da ';': deve cioé essere l'ultimo comando di una istruzione e deve essere seguito da un CR.

Esempi GOTO 120

Fa passare il controllo alla linea 120

GOTO (S > T) * 100 + (S = T) * 200 + (S < T) * 300

Fa proseguire l'esecuzione del programma alla linea 100 200 o 300 a seconda di quale relazione fra S e T è verificata.

GOSUB e RETURN

Il comando GOSUB <expr> è analogo al GOTO eccetto che al momento dell'esecuzione viene salvato il numero della linea e la posizione nella linea in cui si trova il comando; e può essere seguito da ';' e da altri comandi nella stessa linea. Il controllo passa, subito dopo che è stato interpretato, alla linea indicata dall'espressione. L'esecuzione continua dalla nuova linea finché il TBI non incontra il comando RETURN: il controllo torna allora al comando immediatamente successivo all'ultima GOSUB incontrata. Le chiamate a subroutine possono essere annidate e recursive: il massimo livello di annidamento è limitato solo dalle dimensioni dello stack.

FOR e NEXT

Il comando FOR può avere i seguenti formati:

<for> :: = FOR <var> = <expr> TO <expr> <step> <step> :: = STEP <expr>

Esempio FOR X = A + 1 TO 3 * B STEP C − 1

La variabile X (detta variabile di controllo del loop) viene posta al valore dell'espressione seguente A+1. Sono poi calcolate le espressioni che seguono TO e STEP e ne viene salvato il *risultato* (NB il risultato e *non* l'espressione); vengono poi salvati il nome della variabile di controllo (nell'esempio X), il numero della linea e la posizione dentro la linea in cui si trova il comando FOR, poi l'esecuzione prosegue normalmente finché il TBI incontra il comando NEXT <var>. Il valore dell'espressione dopo STEP può essere positivo, negativo o anche zero. Se non viene messa la parola STEP e relativa espressione, il TBI assume per STEP automaticamente il valore +1.

Quando il TBI incontra NEXT < var > controlla che la variabile che segue il NEXT sia la stessa dell'ultimo FOR incontrato. Se non sono uguali, viene eliminato l'ultimo comando FOR e il TBI passa a cercare la variabile del FOR precedente e così via. Quando viene trovato il FOR con la stessa variabile del NEXT, viene sommato alla variabile in questione il valore di STEP specificato nel FOR e salvato in precedenza. Il valore aggiornato della variabile di controllo viene confrontato con quello dell'espressione dopo il TO (anch'esso salvato in precedenza). Se quest'ultimo valore non è ancora stato superato, il controllo torna al comando immediatamente successivo al FOR, se invece il valore dato dal TO è stato superato, il programma continua in sequenza all'istruzione dopo la NEXT.

Se STEP è positivo, si supera il valore di TO quando la variabile del loop è maggiore dell'espressione, se lo STEP è negativo, il TO è superato quando la variabile del loop arriva a valori minori di TO: in questo modo si possono fare dei loop con la variabile di controllo decrescente.

I FOR possono essere annidati: il massimo annidamento possibile è determinato dallo spazio a disposizione dello stack.

Se viene trovato un FOR che ha la stessa variabile di controllo di uno precedente non ancora chiuso, quello precedente viene automaticamente eliminato.

STOP

Questo comando blocca l'esecuzione del programma e fa ritornare il TBI in attesa di comandi. Lo STOP può comparire molte volte in un programma, ma deve sempre essere l'ultimo comando di una istruzione non può cioé essere seguito da ';', ma deve esserci dopo un CR.

Ingresso - Uscita

Nel testo del programma battuto da tastiera, possono comparire liberamente carattere di blank, eccetto che all'interno di numeri, comandi e funzioni. Tutte le parole dei comandi e delle funzioni possono essere abbreviate troncandole e facendole seguire da un punto. Ad esempio P. PR. PRI. PRIN. sono tutte interpretate come PRINT. La parola LET può essere addirittura completamente eliminata. Queste possibilità permettono di risparmiare memoria, poiché bisogna ricordare che le istruzioni vengono messe in memoria come sono battute da tastiera.
Le abbreviazioni più corte sono;

A. = ABS. IF = IF N. = NEXT R. = RND S. = STOP F. = FOR IN. = INPUT P. = PRINT R. = RUN TO=TO GOS.=GOSUB L.=LIST REM= REM S. = SIZE G. = GOTO N. = NEW R. = RETURN S. = STEP

Non c'è possibilità di confusione per il TBI fra abbreviazioni uguali (es. S. per STEP, STOP o SIZE) perché si trovano in contesti diversi. Per bloccare l'esecuzione di un programma o la stampa di un listato si deve battere control-C.

Il dispositivo di uscita (stampante) può essere attivato o disattivato via software con un control-O. Questa possibilità è utile quando si dà in ingresso un nastro perforato e non si vuole l'eco del programma: basta dare il comando NEW per pulire la memoria, poi control-O ed accendere il lettore di nastro. Il programma viene così caricato senza che ne sia fatto l'eco. Alla fine si spegne il lettore e si batte di nuovo control-O per riattivare la stampante.

Per ottenere il nastro perforato conviene seguire questa procedura:

- Battere LIST senza CR
- Accendere il perforatore
- Battere alcuni null (shift-control-P) e un CR
- Terminato il listato, battere altri null e spegnere il perforatore.

Diagnostica

La diagnostica del TBI è ridotta all'essenziale. Quando durante l'esecuzione è rilevato un errore, viene stampata la parola che identifica il tipo di errore e la linea errata con un punto interrogativo inserito nel punto in cui è stato rilevato l'errore. Le parole che identificano i tipi di errore sono 'WHAT?', 'HOW?' e 'SORRY'.
'WHAT?' significa che il TBI non riesce a capire un comando.
Es. WHAT?
20 PT?INT 'QUESTO'
Dove c'è PTINT al posto di PRINT.
'HOW?' significa che il TBI ha capito il comando, ma non sa come eseguirlo.
Es. HOW?
300 LET A = B + C? + 44
Dove B + C provoca overflow.
HOW?
400 GOSUB 567?
dove la linea 567 non esiste.

'SORRY' significa che il TBI ha capito il comando, sa come eseguirlo ma non ha abbastanza memoria a disposizione.

Correzione degli errori

Il TBI si presenta in attesa di comandi con un asterisco. Si possono allora battere comandi diretti o non diretti: la linea viene processata solo quando viene battuto il CR. Se si commette un errore prima di aver chiuso la linea, si può correggere battendo control-H: questo comando elimina l'ultimo carattere battuto: il puntatore torna indietro di un passo ed esce un blank per cancellare sul video eventualmente collegato l'ultimo carattere. Il control-H può essere usato tante volte quante sono necessarie. Se invece si vuole cancellare l'intera linea e non si è ancora battuto il CR finale, bisogna dare un Rubout: il TBI va allora a capo e si presenta con ∧ per indicare che la linea precedente è stata eliminata
Se invece la linea errata è in memoria (cioé è stato battuto il CR), può essere eliminata ribattendo lo stesso numero seguito da un CR.
Se invece si vuole modificare una linea, basta ribatterne il numero e poi la linea esatta, che va a sostituire in memoria quella precedente.
Dopo un'operazione così, il metodo più rapido per verificare la linea modificata è quello di battere LIST seguito dal numero della linea e bloccare subito dopo la stampa con un control-C.

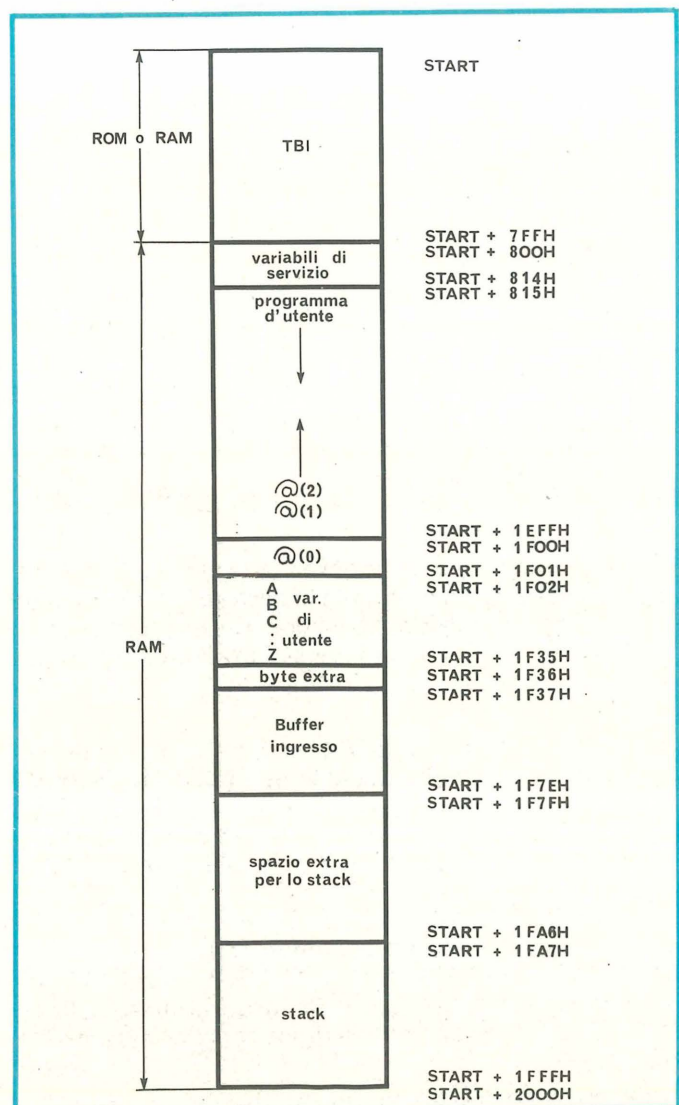



Fig. 1 - Occupazione della memoria.

Rilocazione Interpretatore

Il TBI può essere messo senza troppa difficoltà in qualunque punto della memoria compreso fra 0000H e 77FFH. Con un po' più di lavoro può essere messo nella rimanente memoria. L'occupazione di memoria è mostrata in figura 1. Tutti gli indirizzi sono riferiti a quello iniziale START. Per rilocare il TBI, senza variare le dimensioni della RAM, sono necessarie due operazioni diverse: 1) Bisogna sommare l'indirizzo a cui si vuole rilocare il TBI in tutte le linee di programma che sul listato hanno la R o la dopo il codice operativo, e 2) Bisogna correggere le linee di programma con la X dopo il codice operativo: in questo caso la correzione va fatta prendendo il nuovo valore dell'etichetta e facendo a mano le operazioni indicate sul listato (es. prendere il byte meno significativo, oppure quello più significativo e mettere il bit 7 a uno). Il nuovo valore delle etichette si trova ovviamente sommando al vecchio valore l'indirizzo di partenza che si vuole ottenere. Se ad esempio si vuole rilocare il TBI dall'indirizzo 2080H, si deve sommare 2080H a tutti gli indirizzi segnalati da R o da sul listato. Per rilocare poi, ad esempio, le linee 267 e 268, si somma 2080H al vecchio valore dell'etichetta LIST (letto per esempio nella cross reference table o alla linea 510), ottenendo 22B9H. Si eseguono poi le operazioni indicate: (LIST + 8000H). SHR. 8 significa prendere il byte più significativo dopo aver sommato 8000H, ottenendo quindi per la linea 267 il valore A2H, mentre LIST&OFFH significa prendere il byte meno significativo, ottenendo per la linea 268 il valore B9H.

Per variare lo spazio a disposizione del programma di utente, bisogna invece cambiare gli indirizzi di tutto quanto è dopo il programma di utente, cioé (vedi fig. 1) delle variabili di utente, del buffer di ingresso e dello spazio per lo stack. I punti del programma che devono essere modificati in questo caso sono tutti indicati da , inoltre anche le linee 1298 e 1301 devono essere modificate in questo caso, compiendo però l'operazione indicata (per questo motivo c'è la X). Per far funzionare invece il programma nei secondi 32 Kbyte, bisogna trasformare tutti i caratteri ASCII in modo che abbiano il bit 7 a uno (anche quelli che arrivano da tastiera) e modificare la routine EXEC in modo che riconosca gli indirizzi segnalati dal bit più significativo a zero (vedere anche la spiegazione sul listato). Bisogna poi fare la rilocazione normale come detto in precedenza.

Porte di ingresso-uscita

Le routine di ingresso-uscita sono scritte per una UART con una porta per lo stato e una per i dati, agli indirizzi UARST e UARDAT: dovendo adattare il programma al proprio sistema, bisogna cambiare questi indirizzi che compaiono nelle linee 1589, 1600, 1585, 1596 e che sul listato sono DFH e DEH perché tali erano sul mio sistema. Discorso analogo vale per le maschere dei bit di stato di ingresso e di uscita chiamati FLAGIN e FLAOUT: dalla cross reference table si ricava che compaiono nelle linee 1589 e 1586.

Bibliografia

Dr. Dobb's Journal of Computer Calisthenics & Orthodontia (maggio '76)

```
                              TINY.BASIC      21$GIUGNO$1979      PAGE   1
 LOC   OBJ CODE M STMT SOURCE STATEMENT                                BAU 4.1

                    1 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                    2 ;                                                                       ;
                    3 ;                                                                       ;
                    4 ;................................MAIN.....................................;
                    5 ;                                                                       ;
                    6 ;                                                                       ;
                    7 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                    8 ;                                                                       ;
                    9 ;                                                                       ;
                   10 ; MAIN E' LA ROUTINE CHE RACCOGLIE IL PROGRAMMA D'UTENTE E LO METTE IN MEMORIA ;
                   11 ;                                                                       ;
                   12 ; ALLA PARTENZA STAMPA '<CR> OK <CR>' ED INIZIALIZZA LO STACK POINTER E ALCUNE ;
                   13 ; ALTRE VARIABILI. POI STAMPA '*' E LEGGE UNA LINEA. SE LA LINEA COMINCIA  CON ;
                   14 ; UN NUMERO DIVERSO DA ZERO, ESSO E' INTERPRETATO COME NUMERO DELLA LINEA,   ;
                   15 ; VIENE CONVERTITO IN BINARIO SU 16 BIT,E VIENE MESSO IN MEMORIA CON IL RESTO ;
                   16 ; DELLA LINEA (COMPRESO IL CR FINALE).SE IN MEMORIA E' GIA' PRESENTE UNA LINEA ;
                   17 ; CON LO STESSO NUMERO,QUESTA VIENE CANCELLATA E SOSTITUITA CON L'ULTIMA LINEA ;
                   18 ; BATTUTA.SE LA LINEA E' COMPOSTA SOLO DAL NUMERO E DAL CR,NON VIENE MESSA IN ;
                   19 ; MEMORIA,MA VIENE SOLO CANCELLATA L'EVENTUALE LINEA GIA' PRESENTE CON LO STES-;
                   20 ; SO NUMERO.                                                            ;
                   21 ;                                                                       ;
                   22 ; DOPO CHE HA INSERITO,CANCELLATO O SOSTITUITO UNA LINEA,IL PROGRAMMA TORNA DA ;
                   23 ; CAPO AD ATTENDERE UN'ALTRA LINEA DI PROGRAMMA.IL PROGRAMMA ESCE DA QUESTO  ;
                   24 ; LOOP DI ATTESA QUANDO INCONTRA UNA LINEA CON NUMERO ZERO O SENZA NUMERO E IL ;
                   25 ; CONTROLLO PASSA ALLA SEZIONE 'DIRECT'                                 ;
                   26 ;                                                                       ;
                   27 ; LA SAVEAREA,CIOE' L'AREA IN CUI E' MEMORIZZATO IL PROGRAMMA D'UTENTE, COMIN- ;
                   28 ; CIA ALLA LOCAZIONE CHIAMATA 'TXTBGN' E TERMINA A 'TXTEND'. IL PROGRAMMA INI- ;
                   29 ; ZIA SEMPRE A RIEMPIRE LA MEMORIA DA 'TXTBGN'. LA PARTE DI SAVEAREA ANCORA ;
                   30 ; VUOTA E' PUNTATA DALLA VARIABILE 'TXTUNF'.                            ;
                   31 ; LA VARIABILE 'CURRNT' PUNTA AL NUMERO DELLA LINEA CHE IL BASIC STA INTERPRE- ;
                   32 ; TANDO. DURANTE LA ROUTINE MAIN OPPURE L'ESECUZIONE DI UN COMANDO DIRETTO ;
                   33 ; 'CURRNT' PUNTA AD UNA LINEA FITTIZIA CHE COMINCIA PER ZERO.           ;
                   34 ;                                                                       ;
                   35 ; NOTA DI LETTURA : IL SIMBOLO  ->  VA LETTO 'PUNTA'                    ;
                   36 ;                                                                       ;
                   37 ;                                                                       ;
                   38 ;                                                                       ;
0000  F3           39 START    DI
0001  310020   @   40          LD SP,STACK            ;INIZIALIZZA LO STACK POINTER
0004  00           41          NOP                    ;SPAZIO PER EVENTUALE CHIAMATA
0005  00           42          NOP                    ;ROUTINE DI INIZIALIZZAZIONE
0006  00           43          NOP                    ;DEL SISTEMA
0007  CD7507   R   44 ST1      CALL CRLF
000A  111401   R   45          LD DE,OK               ;CARICA IL PUNT. A 'OK'
000D  97           46          SUB A                  ;A=0
000E  CD8106   R   47          CALL PRTSTG            ;STAMPA LA STRINGA FINO A CR
0011  211800   R   48          LD HL,ST2+1            ;CURRNT -> ST2+1 CIOE'
0014  220108   R   49          LD (CURRNT),HL         ;AD UNA LINEA CON NUM. = 0
0017  210000       50 ST2      LD HL,0
001A  220708   R   51          LD (LOPVAR),HL         ;INIZIALIZZA PARAMETRI
001D  220308   R   52          LD (STKGOS),HL         ;PER LE ISTRUZ. 'FOR' E 'GOSUB'
0020  3E2A         53 ST3      LD   A,'*'             ;STAMPA * E
0022  CD0E06   R   54          CALL  GETLN            ;LEGGE UNA LINEA
0025  D5           55          PUSH  DE               ;DE -> ALLA FINE DELLA LINEA
0026  11371F   @   56          LD DE,BUFFER           ;DE -> ALL'INIZ. DELLA LINEA
0029  CDDD00   R   57          CALL  TSTNUM           ;PROVA SE COMINCIA CON UN NUMERO
002C  CD8600   R   58          CALL  IGNBLK           ;IGNORA I BLANK NON SIGNIFICATIVI
                              TINY.BASIC      21$GIUGNO$1979      PAGE   2
 LOC   OBJ CODE M STMT SOURCE STATEMENT                                BAU 4.1

002F  7C           59          LD A,H                 ;HL=VALORE DEL NUMERO OPPURE
0030  B5           60          OR L                   ;0 SE NON C'E' ALCUN NUMERO
0031  C1           61          POP BC                 ;BC -> ALLA FINE DELLA LINEA
0032  CAC701   R   62          JP Z,DIRECT            ;HL=0 , E' UN COMANDO
0035  1B           63          DEC DE                 ;DE -> ORA AL BYTE PRECEDENTE
0036  7C           64          LD A,H                 ;L'INIZ. DELL'ISTRUZIONE :
0037  12           65          LD (DE),A              ;QUI VIENE SALVATO IL NUMERO
0038  1B           66          DEC DE                 ;DELLA LINEA (SU 2 BYTE)
0039  7D           67          LD A,L
003A  12           68          LD (DE),A
003B  C5           69          PUSH BC                ;BC -> ALLA FINE DELLA LINEA
003C  D5           70          PUSH DE                ;DE -> ALL'INIZIO DELLA LINEA
003D  79           71          LD A,C                 ;VIENE CALCOLATA LA LUNGHEZZA
003E  93           72          SUB E                  ;DELLA LINEA IN BYTE :
003F  F5           73          PUSH AF                ;IL RISULTATO SI TROVA IN A
0040  CD5706   R   74          CALL FNDLN             ;CERCA IN SAVEAREA SE C'E' GIA'
0043  D5           75          PUSH DE                ;DE -> A SAVEAREA
0044  C25700   R   76          JP NZ,ST4              ;NZ:NON E' STATA TROVATA:INSERISCE
0047  D5           77          PUSH DE                ;Z:TROVATA LA CANCELLA
0048  CD7506   R   78          CALL FNDNXT            ;CERCA LA LINEA SUCCESSIVA
                   79                                 ;DE -> ALLA LINEA SUCCESSIVA
004B  C1           80          POP BC                 ;BC -> ALLA LINEA DA CANCELLARE
004C  2A1308   R   81          LD HL,(TXTUNF)         ;HL -> ALLA SAVEAREA VUOTA
004F  CD1607   R   82          CALL MVUP              ;TIRA SU LE LINEE SUCCESSIVE
                   83                                 ;ELIMINANDO QUELLA DA CANCELLARE
0052  60           84          LD H,B                 ;AGGIORNA TXTUNF CHE DEVE
0053  69           85          LD L,C                 ;PUNTARE ALLA PARTE VUOTA
0054  221308   R   86          LD (TXTUNF),HL         ;DELLA SAVEAREA
0057  C1           87 ST4      POP BC                 ;SI PREPARA AD INSERIRE
```

```
00CD  BE          173          CP (HL)                ;CONFRONTA IL BYTE SUCCESSIVO ALLA
00CE  23          174 TC1      INC HL                 ;CALL CON IL TESTO PUNTATO DA DE
00CF  CAD900   R  175          JP Z,TC2               ;SE SONO DIVERSI AGGIUNGE IL
00D2  C5          176          PUSH BC                ;SECONDO BYTE DOPO LA CALL AL
00D3  4E          177          LD C,(HL)              ;VECCHIO PC, FA CIOE' UN SALTO
00D4  0606        178          LD B,0                 ;RELATIVO SE I 2 BYTE SONO DIVERSI
00D6  09          179          ADD HL,BC
00D7  C1          180          POP BC
00D8  1B          181          DEC DE
00D9  13          182 TC2      INC DE                 ;SE SONO UGUALI RITORNA
00DA  23          183          INC HL                 ;AL TERZO BYTE DOPO LA 'CALL TSTC'
00DB  E3          184          EX (SP),HL
00DC  C9          185          RET
                  186 ;
00DD  210000      187 TSTNUM   LD HL,0                ;CONTROLLA SE IL TESTO
00E0  44          188          LD B,H                 ;E' UN NUMERO IN ASCII
00E1  CD8600   R  189          CALL IGNBLK
00E4  FE30        190 TN1      CP '0'                 ;SE NO TORNA CON ZERO
00E6  D8          191          RET C                  ;IN B ED IN HL
00E7  FE3A        192          CP 3AH                 ;SE LO E',LO CONVERTE
00E9  D0          193          RET NC                 ;IN BINARIO IN HL E METTE IN B
00EA  3EF0        194          LD A,0F0H              ;IL NUMERO DI CIFRE DEL NUMERO
00EC  A4          195          AND H                  ;IL NUMERO PRIMA DI ESSERE
00ED  C20701   R  196          JP NZ,QHOW             ;MOLTIPLICATO PER 10 DEVE
00F0  04          197          INC B                  ;ESSERE MINORE DI 4096
00F1  C5          198          PUSH BC                ;ALTRIMENTI C'E' OVERFLOW
00F2  44          199          LD B,H                 ;HL=HL*10+(NUOVA CIFRA)
00F3  4D          200          LD C,L                 ;LA MOLTIPLICAZIONE *10 E'
00F4  29          201          ADD HL,HL              ;FATTA CON SOMME E SHIFT
00F5  29          202          ADD HL,HL
00F6  09          203          ADD HL,BC
00F7  29          204          ADD HL,HL
00F8  1A          205          LD A,(DE)              ;(NUOVA CIFRA) E' RICAVATA
00F9  13          206          INC DE                 ;DAL CODICE ASCII
00FA  D630        207          SUB '0'                ;IN ASCII 'CIFRA'=30H+CIFRA=
00FC  85          208          ADD A,L                ;                  ='0'+CIFRA
00FD  6F          209          LD L,A
00FE  3E00        210          LD A,00
0100  8C          211          ADC A,H
0101  67          212          LD H,A
0102  C1          213          POP BC                 ;QUESTO E' FATTO PER
0103  1A          214          LD A,(DE)              ;OGNI CIFRA
0104  F2E400   R  215          JP P,TN1               ;IL FLAG DI SEGNO ATTIVATO
                  216                                 ;INDICA OVERFLOW
                  217 ;                                                                       ;
0107  D5          218 QHOW     PUSH DE                ;PER QUESTE TRE ISTRUZIONI
0108  110E01   R  219 AHOW     LD DE,HOW              ;VEDERE LA ROUTINE 'ERROR'
010B  C3DC05   R  220          JP ERROR
010E  484F5720    221 HOW      DEFM 'HOW ?'           ;MESSAGGI VARI
0113  0D          222          DEFB CR
0114  4F4B        223 OK       DEFM 'OK'
0116  0D          224          DEFB CR

                              TINY.BASIC      21$GIUGNO$1979      PAGE   5
 LOC   OBJ CODE M STMT SOURCE STATEMENT                                BAU 4.1

0117  57484154    225 WHAT     DEFM 'WHAT ?'
011D  0D          226          DEFB CR
011E  534F5252    227 SORRY    DEFM 'SORRY'
0123  0D          228          DEFB CR
                  229 ;                                                                       ;
                  230 ;                                                                       ;
                  231 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  232 ;                                                                       ;
                  233 ;                                                                       ;
                  234 ;..............TABELLE...............DIRECT..............EXEC...................;
                  235 ;                                                                       ;
                  236 ;                                                                       ;
                  237 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  238 ;                                                                       ;
                  239 ;                                                                       ;
                  240 ; QUESTA PARTE DI PROGRAMMA CERCA UNA STRINGA IN UNA TABELLA, E, QUANDO L'HA ;
                  241 ; TROVATA, SALTA ALL'INDIRIZZO SPECIFICATO IN TABELLA.                  ;
                  242 ;                                                                       ;
                  243 ; IN 'EXEC' DE DEVE PUNTARE ALLA STRINGA E HL DEVE PUNTARE A TABELLA-1  ;
                  244 ; IN 'DIRECT' DE DEVE PUNTARE ALLA STRINGA E HL A TAB1-1,POICHE' TAB1 E' LA ;
                  245 ; TABELLA DEI COMANDI DIRETTI.                                         ;
                  246 ;                                                                       ;
                  247 ; SE VIENE TROVATO UN '.' NELLA STRINGA, IL CONFRONTO HA TERMINE E SE E' STA- ;
                  248 ; TA TROVATA UNA PARTE DELLA STRINGA, SI PROCEDE COME SE FOSSE STATA TROVATA ;
                  249 ; TUTTA LA STRINGA. AD ESEMPIO 'P.' , 'PR.' , 'PRT.' , 'PRIN.' O 'PRINT' SONO ;
                  250 ; TUTTE ACCOPPIATE ALLA STRINGA 'PRINT' IN TABELLA : QUESTO PERMETTE L'USO DI ;
                  251 ; COMANDI ABBREVIATI CHE FANNO RISPARMIARE MEMORIA.                     ;
                  252 ;                                                                       ;
                  253 ; LA TABELLA CONSISTE DI UNA SERIE (ESPANDIBILE) DI VOCI. OGNI VOCE E' COMPO- ;
                  254 ; STA DA UNA STRINGA DI CARATTERI COL BIT 7 A ZERO, E UN INDIRIZZO DI SALTO ;
                  255 ; MEMORIZZATO COME BYTE ALTO-BYTE BASSO (AL CONTRARIO CIOE' DI QUELLO CHE FA- ;
                  256 ; REBBE IL MICROPROCESSORE) CON IL BIT 7 DEL BYTE ALTO A 1 : QUESTO SERVE A ;
                  257 ; DISTINGUERE LA FINE DEI CARATTERI E VINCOLA LA RILOCABILITA' DELL'INTERPRET- ;
                  258 ; TATORE CHE DEVE STARE NEI PRIMI 32 KBYTE DI MEMORIA.                  ;
                  259 ;                                                                       ;
                  260 ; LA FINE DELLA TABELLA E' COSTITUITA DA UNA VOCE SENZA STRINGA, CIOE' SOLO ;
```

```
0058  2A1308    R  88         LD HL,(TXTUNF)      ;MA PRIMA CONTROLLA SE
005B  F1           89         POP AF              ;LA LUNGHEZZA DELLA NUOVA
005C  E5           90         PUSH HL             ;LINEA E' 3 (NUM, LINEA E CR)
005D  FE03         91         CP 3                ;SE E' COSI' NON LA INSERISCE
005F  CA0000    R  92         JP Z,START          ;DEVE PULIRE LO STACK
0062  85           93         ADD A,L             ;CALCOLA IL NUOVO VALORE
0063  6F           94         LD L,A              ;DEL PUNTATORE TXTUNF
0064  3E00         95         LD A,0
0066  8C           96         ADC A,H
0067  67           97         LD H,A              ;HL -> ALLA NUOVA AREA VUOTA
0068  11001F    @  98         LD DE,TXTEND        ;CONTROLLA SE C'E' ANCORA
006B  CD8000    R  99         CALL COMP           ;ABBASTANZA MEMORIA
006E  D20706    R 100         JP NC,QSORRY        ;PECCATO! NON CE N'E' PIU'!
0071  221308    R 101         LD (TXTUNF),HL      ;OK,AGGIORNA TXTUNF
0074  D1          102         POP DE              ;DE -> ALLA VECCHIA AREA VUOTA
0075  CD2107    R 103         CALL MVDOWN         ;FA SPAZIO IN SAVEAREA
                  104                             ;PER INSERIRE LA NUOVA LINEA
0078  D1          105         POP DE              ;DE -> ALL'INIZ. DELLA NUOVA LINEA
0079  E1          106         POP HL              ;HL -> ALLA FINE DELLA NUOVA LINEA
007A  CD1607    R 107         CALL MVUP           ;SPOSTA LA NUOVA LINEA IN SAVEAREA
007D  C32000    R 108         JP ST3              ;PUNTATA DA BC
                    TINY.BASIC     21$GIUGNO$1979       PAGE   3
LOC   OBJ CODE M STMT SOURCE STATEMENT                   BAU 4.1

                  109  *E
                  110  ;                                                                     ;
                  111  ;                                                                     ;
                  112  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  113  ;                                                                     ;
                  114  ;                                                                     ;
                  115  ;..........COMP............IGNBLK...........TSTC...........TSTNUM.........;
                  116  ;                                                                     ;
                  117  ;                                                                     ;
                  118  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  119  ;                                                                     ;
                  120  ;                                                                     ;
                  121  ; ROUTINE DI SERVIZIO : IL LORO FUNZIONAMENTO E' SPIEGATO NEI COMMENTI ;
                  122  ;                                                                     ;
                  123  ;                                                                     ;
0080  7C          124  COMP   LD A,H              ;ROUTINE DI CONFRONTO FRA HL E DE
0081  BA          125         CP D                ;CONFRONTA HL E DE
0082  C0          126         RET NZ              ;RITORNA CON I FLAG DI CARRY E
0083  7D          127         LD A,L              ;DI ZERO POSIZIONATI CORRETTAMENTE
0084  BB          128         CP E                ;MA DISTRUGGE IL CONTENUTO
0085  C9          129         RET                 ;DELL'ACCUMULATORE
                  130  ;                                                                     ;
0086  1A          131  IGNBLK LD A,(DE)           ;ROUTINE CHE IGNORA I BLANK
0087  FE20        132         CP ' '              ;NEL TESTO PUNTATO DA DE
0089  C0          133         RET NZ              ;TORNA COL PRIMO CARATTERE
008A  13          134         INC DE              ;NON BLANK NELL'ACCUMULATORE
008B  C38600    R 135         JP IGNBLK           ;E CON DE AGGIORNATO
                  136  ;                                                                     ;
008E  F1          137  FINISH POP AF              ;ROUTINE DI RICONOSCIMENTO
008F  CDBF05    R 138         CALL FIN            ;DI FINE COMANDO
0092  C3D805    R 139         JP QWHAT            ;STAMPA 'WHAT?' SE E' SBAGLIATO
                  140  ;                                                                     ;
0095  CD8600    R 141  TSTV   CALL IGNBLK         ;ROUTINE RICONOSCIMENTO VARIABILI
0098  D640        142         SUB '@'             ;CONTROLLA SE E' UNA VARIABILE
009A  D8          143         RET C               ;CARRY:NON E' UNA VARIABILE
009B  C2B900    R 144         JP NZ,TV1           ;NON E' IL VETTORE '@'
009E  13          145         INC DE              ;E' IL VETTORE '@'
009F  CD1905    R 146         CALL PARN           ;@ DEVE ESSERE SEGUITO
00A2  29          147         ADD HL,HL           ;DA (<EXPR>) COME INDICE
00A3  DA0701    R 148         JP C,QHOW           ;L'INDICE E' TROPPO GRANDE E
00A6  D5          149         PUSH DE             ;@ VA A SPORCARE IL PROGRAMMA
00A7  EB          150         EX DE,HL            ;DI UTENTE?
00A8  CD6305    R 151         CALL SIZE           ;CERCA QUANTA MEMORIA C'E'
00AB  CD8000    R 152         CALL COMP           ;LIBERA
00AE  DA0806    R 153         JP C,ASORRY         ;NON C'E' MEM.: STAMPA 'SORRY'
00B1  21001F    @ 154         LD HL,VARBGN        ;SE NO FORNISCE L'INDIRIZZO
00B4  CD8605    R 155         CALL SUBDE          ;DI @(<EXPR>) E LO
00B7  D1          156         POP DE              ;METTE IN HL
00B8  C9          157         RET                 ;IL FLAG C E' AZZERATO
00B9  FE1B        158  TV1    CP 'Z'-'@'+1        ;NON E' @, ALLORA E' FRA A E Z?
00BB  3F          159         CCF                 ;SE NO ATTIVA IL CARRY
00BC  D8          160         RET C
00BD  13          161         INC DE              ;SE E' FRA A E Z CALCOLA
00BE  21001F    @ 162         LD HL,VARBGN        ;L'INDIRIZZO DELLA VARIABILE
00C1  07          163         RLCA                ;E LO METTE IN HL COL CARRY=0
00C2  85          164         ADD A,L
00C3  6F          165         LD L,A
00C4  3E00        166         LD A,0

                    TINY.BASIC     21$GIUGNO$1979       PAGE   4
LOC   OBJ CODE M STMT SOURCE STATEMENT                   BAU 4.1

00C6  8C          167         ADC A,H
00C7  67          168         LD H,A
00C8  C9          169         RET
                  170  ;                                                                     ;
00C9  E3          171  TSTC   EX (SP),HL          ;ROUTINE DI CONFRONTO CARATTERI
00CA  CD8600    R 172         CALL IGNBLK         ;IGNORA I BLANK
```

```
                  261  ; DA UN INDIRIZZO DI SALTO, A CUI SI VA SE NON SI RIESCE A TROVARE LA STRINGA ;
                  262  ; NELLA TABELLA                                                      ;
                  263  ;                                                                     ;
                  264  ;                                                                     ;
                  265  TAB1                               ;TABELLE COMANDI DIRETTI
0124  4C495354    266         DEFM 'LIST'                 ;NOME DEL COMANDO
0128  82        X 267         DEFB (LIST+8000H).SHR.8     ;INDIRIZZO ROUTINE CHE ESEGUE
0129  39        X 268         DEFB LIST&0FFH              ;IL COMANDO. PRIMA C'E' IL
012A  52554E      269         DEFM 'RUN'                  ;BYTE PIU' SIGNIFICATIVO COL BIT
012D  82        X 270         DEFB (RUN+8000H).SHR.8      ;7 A 1,POI C'E' IL BYTE
012E  09        X 271         DEFB RUN&0FFH               ;MENO SIGNIFICATIVO
012F  4E4557      272         DEFM 'NEW'
0132  81        X 273         DEFB (NEW+8000H).SHR.8
0133  FA        X 274         DEFB NEW&0FFH
                  275  TAB2                               ;TABELLA COMANDI DIRETTI E NON
0134  4E455854    276         DEFM 'NEXT'
0138  83        X 277         DEFB (NEXT+8000H).SHR.8
0139  39        X 278         DEFB NEXT&0FFH
013A  4C4554      279         DEFM 'LET'
013D  84        X 280         DEFB (LET+8000H).SHR.8
013E  0F        X 281         DEFB LET&0FFH
013F  4946        282         DEFM 'IF'

                    TINY.BASIC     21$GIUGNO$1979       PAGE   6
LOC   OBJ CODE M STMT SOURCE STATEMENT                   BAU 4.1

0141  83        X 283         DEFB (IF+8000H).SHR.8
0142  94        X 284         DEFB IF&0FFH
0143  474F544F    285         DEFM 'GOTO'
0147  82        X 286         DEFB (GOTO+8000H).SHR.8
0148  28        X 287         DEFB GOTO&0FFH
0149  474F5355    288         DEFM 'GOSUB'
014E  82        X 289         DEFB (GOSUB+8000H).SHR.8
014F  97        X 290         DEFB GOSUB&0FFH
0150  52455455    291         DEFM 'RETURN'
0156  82        X 292         DEFB (RETURN+8000H).SHR.8
0157  B9        X 293         DEFB RETURN&0FFH
0158  52454D      294         DEFM 'REM'
015B  83        X 295         DEFB (REM+8000H).SHR.8
015C  8E        X 296         DEFB REM&0FFH
015D  464F52      297         DEFM 'FOR'
0160  82        X 298         DEFB (FOR+8000H).SHR.8
0161  D4        X 299         DEFB FOR&0FFH
0162  494E5055    300         DEFM 'INPUT'
0167  83        X 301         DEFB (INPUT+8000H).SHR.8
0168  AF        X 302         DEFB INPUT&0FFH
0169  5052494E    303         DEFM 'PRINT'
016E  82        X 304         DEFB (PRINT+8000H).SHR.8
016F  51        X 305         DEFB PRINT&0FFH
0170  53544F50    306         DEFM 'STOP'
0174  82        X 307         DEFB (STOP+8000H).SHR.8
0175  03        X 308         DEFB STOP&0FFH
0176  84        X 309         DEFB (DEFLT+8000H).SHR.8
0177  09        X 310         DEFB DEFLT&0FFH
0178  42595445    311         DEFM 'BYTE LIBERI'          ;PER ALTRE ISTRUZIONI
                  312  ;
                  313  ;
                  314  TAB4                               ;TABELLA DELLE FUNZIONI
0183  524E44      315         DEFM 'RND'
0186  85        X 316         DEFB (RND+8000H).SHR.8
0187  2A        X 317         DEFB RND&0FFH
0188  414253      318         DEFM 'ABS'
018B  85        X 319         DEFB (ABS+8000H).SHR.8
018C  57        X 320         DEFB ABS&0FFH
018D  53495A45    321         DEFM 'SIZE'
0191  85        X 322         DEFB (SIZE+8000H).SHR.8
0192  63        X 323         DEFB SIZE&0FFH
0193  85        X 324         DEFB (XP40+8000H).SHR.8
0194  08        X 325         DEFB XP40&0FFH
0195  5350415A    326         DEFM 'SPAZIO  LIBERO'       ;PER ALTRE FUNZIONI
                  327  ;
                  328  TAB5                               ;'TO' NELL'ISTRUZ. 'FOR'
01A3  544F        329         DEFM 'TO'
01A5  82        X 330         DEFB (FR1+8000H).SHR.8
01A6  E4        X 331         DEFB FR1&0FFH
01A7  85        X 332         DEFB (QWHAT+8000H).SHR.8
01A8  D8        X 333         DEFB QWHAT&0FFH
                  334  TAB6                               ;'STEP' NELL'ISTRUZ. 'FOR'
01A9  53544550    335         DEFM 'STEP'
01AD  82        X 336         DEFB (FR2+8000H).SHR.8
01AE  F0        X 337         DEFB FR2&0FFH
01AF  82        X 338         DEFB (FR3+8000H).SHR.8
01B0  F6        X 339         DEFB FR3&0FFH
                  340  ;

                    TINY.BASIC     21$GIUGNO$1979       PAGE   7
LOC   OBJ CODE M STMT SOURCE STATEMENT                   BAU 4.1

                  341  TAB8                               ;OPERATORI DI RELAZIONE
01B1  3E3D        342         DEFM '>='
01B3  84        X 343         DEFB (XP11+8000H).SHR.8
01B4  27        X 344         DEFB XP11&0FFH
```

```
01B5   23          345          DEFB '#'
01B6   84       X  346          DEFB (XP12+8000H).SHR.8
01B7   2D       X  347          DEFB XP12&0FFH
01B8   3E          348          DEFB '>'
01B9   84       X  349          DEFB (XP13+8000H).SHR.8
01BA   33       X  350          DEFB XP13&0FFH
01BB   3D          351          DEFB '='
01BC   84       X  352          DEFB (XP15+8000H).SHR.8
01BD   42       X  353          DEFB XP15&0FFH
01BE   3C3D        354          DEFM '<='
01C0   84       X  355          DEFB (XP14+8000H).SHR.8
01C1   3A       X  356          DEFB XP14&0FFH
01C2   3C          357          DEFB '<'
01C3   84       X  358          DEFB (XP16+8000H).SHR.8
01C4   48       X  359          DEFB XP16&0FFH
01C5   84       X  360          DEFB (XP17+8000H).SHR.8
01C6   4E       X  361          DEFB XP17&0FFH
                   362  ;
                   363  ;
                   364  ;
01C7   212301   R  365  DIRECT  LD HL,TAB1-1          ;ROUTINE PER L'ESECUZIONE DEI
                   366                                ;COMANDI DIRETTI
                   367  EXEC
01CA   CD8600   R  368  EX0     CALL IGNBLK           ;IGNORA I BLANK NON SIGNIFICATIVI
01CD   D5          369          PUSH DE               ;SALVA IL PUNTATORE
01CE   1A          370  EX1     LD A,(DE)             ;SE VIENE TROVATO UN '.'
01CF   13          371          INC DE                ;NELLA STRINGA PRIMA DI UNA
01D0   FE2E        372          CP '.'                ;QUALCHE DIFFERENZA, LA STRINGA
01D2   CAEB01   R  373          JP Z,EX3              ;VIENE CONSIDERATA TROVATA
01D5   23          374          INC HL                ;HL -> TABELLA
01D6   BE          375          CP (HL)               ;SE SONO UGUALI, SI CONTINUA
01D7   CACE01   R  376          JP Z,EX1              ;IL CONFRONTO
01DA   3E7F        377          LD A,7FH              ;SE NO SI CONTROLLA SE IL BIT 7
01DC   1B          378          DEC DE                ;IN TABELLA E' ALTO:IN TAL CASO
01DD   BE          379          CP (HL)               ;E' L'INDIRIZZO DI SALTO
01DE   DAF201   R  380          JP C,EX5              ;SE NO,CONTINUA LA RICERCA
01E1   23          381  EX2     INC HL
01E2   BE          382          CP (HL)
01E3   D2E101   R  383          JP NC,EX2
01E6   23          384          INC HL                ;VA ALLA VOCE SUCCESSIVA IN TAB.
01E7   D1          385          POP DE                ;RECUPERA IL PUNTATORE A STRINGA
01E8   C3CA01   R  386          JP EX0                ;PROVA CON UN'ALTRA VOCE IN TAB.
01EB   3E7F        387  EX3     LD A,7FH              ;TROVATO UN ACCORDO PARZIALE
                   388                                ;FRA STRINGA E VOCE IN TABELLA
01ED   23          389  EX4     INC HL                ;VA FINO ALL'INDIR. DI SALTO
01EE   BE          390          CP (HL)               ;SEGNALATO DAL BIT 7 ALTO
01EF   D2ED01   R  391          JP NC,EX4
01F2   7E          392  EX5     LD A,(HL)             ;CARICA HL CON L'INDIRIZZO DI
01F3   23          393          INC HL                ;SALTO LETTO IN TABELLA
01F4   6E          394          LD L,(HL)
01F5   E67F        395          AND 7FH               ;MASCHERA IL BIT 7
01F7   67          396          LD H,A
01F8   F1          397          POP AF                ;TOGLIE DI MEZZO LA PORCHERIA
01F9   E9          398          JP (HL)               ;VA AD ESEGUIRE IL COMANDO

              TINY.BASIC      21$GIUGNO$1979      PAGE    8
LOC    OBJ CODE M STMT SOURCE STATEMENT               BAU 4.1

                   399  ;                                                                        ;
                   400  ;                                                                        ;
                   401  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   402  ;                                                                        ;
                   403  ;                                                                        ;
                   404  ; QUELLO CHE SEGUE E' IL SEGMENTO CHE ESEGUE I COMANDI DIRETTI E LE ISTRUZIO-;
                   405  ; NI. IL CONTROLLO PASSA ALLE SEZIONI DI PROGRAMMA INDIVIDUATE ATTRAVERSO LE ;
                   406  ; TABELLE 'DIRECT' ED 'EXEC' PRECEDENTI. DOPO CHE IL COMANDO E' STATO ESEGUITO,;
                   407  ; IL CONTROLLO VIENE CEDUTO ALLE SEGUENTI SEZIONI :                      ;
                   408  ; PER 'LIST','NEW','STOP' TORNA A START                                  ;
                   409  ; PER 'RUN' VA AD ESEGUIRE LA PRIMA LINEA MEMORIZZATA SE E' PRESENTE,SE NO VA ;
                   410  ; A START                                                                ;
                   411  ; PER 'GOTO' E 'GOSUB' VA AD ESEGUIRE LA LINEA INDICATA                  ;
                   412  ; PER 'RETURN' E 'NEXT' TORNA INDIETRO ALLA LINEA DI RITORNO SALVATA PRIMA ;
                   413  ; PER TUTTI GLI ALTRI COMANDI,PROCEDE IN SEQUENZA A MENO CHE 'CURRNT' NON ;
                   414  ; PUNTI A UNA LINEA COMINCIANTE PER ZERO : IN TAL CASO VA A START        ;
                   415  ;                                                                        ;
                   416  ;                                                                        ;
                   417  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   418  ;                                                                        ;
                   419  ;                                                                        ;
                   420  ;..........NEW............STOP............RUN.(ET.ALII)......GOTO..............;
                   421  ;                                                                        ;
                   422  ;                                                                        ;
                   423  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   424  ;                                                                        ;
                   425  ;                                                                        ;
                   426  ; 'NEW<CR>' PONE 'TXTUNF' COME PUNTATORE A 'TXTBGN'                      ;
                   427  ;                                                                        ;
                   428  ; 'STOP<CR>' TORNA INDIETRO A START                                      ;
                   429  ;                                                                        ;
                   430  ; 'RUN<CR>' CERCA LA PRIMA LINEA MEMORIZZATA , METTE IL SUO INDIRIZZO NELLA ;
                   431  ; LOCAZIONE 'CURRNT'  E LA ESEGUE (SOLO I COMANDI DI TAB2 SONO VALIDI PER ;
                   432  ; IL PROGRAMMA)                                                          ;
```

```
                   518  ;
0251   0E06        519  PRINT   LD C,6                ;C= NUMERO DI SPAZI (DEFAULT 6)
0253   CDC900   R  520          CALL TSTC             ;SE NON CI SONO ARGOMENTI
0256   3B          521          DEFB ';'              ;E C'E' UN ';'
0257   06          522          DEFB (PR2-$-1)&0FFH
0258   CD7507   R  523          CALL CRLF             ;STAMPA UN CR-LF
025B   C31F02   R  524          JP RUNSML             ;E PROSEGUE SULLA STESSA LINEA
025E   CDC900   R  525  PR2     CALL TSTC             ;SE NON CI SONO ARGOMENTI
0261   0D          526          DEFB CR               ;E C'E' UN CARRIAGE-RETURN
0262   06          527          DEFB (PR0-$-1)&0FFH
0263   CD7507   R  528          CALL CRLF             ;STAMPA UN CR-LF E
0266   C30F02   R  529          JP RUNNXL             ;VA AD ESEGUIRE LA LINEA SUCCESSIVA
0269   CDC900   R  530  PR0     CALL TSTC             ;C'E' UN FORMAT?
026C   23          531          DEFB '#'
026D   07          532          DEFB (PR1-$-1)&0FFH
026E   CD1D04   R  533          CALL EXPR             ;SI';VALUTA L'ESPRESSIONE
0271   4D          534          LD C,L                ;SALVA IL RISULTATO IN C
0272   C37B02   R  535          JP PR3                ;GUARDA SE CI SONO PIU' COSE DA STAMPARE
0275   CD8F06   R  536  PR1     CALL QTSTG            ;C'E' UNA STRINGA DA STAMPARE?
0278   C38C02   R  537          JP PR8                ;SE NO, DEVE ESSERE UN'ESPRESSIONE
027B   CDC900   R  538  PR3     CALL TSTC             ;SE C'E' UNA ',' CERCA
027E   2C          539          DEFB ','
027F   06          540          DEFB (PR6-$-1)&0FFH
0280   CDBF05   R  541          CALL FIN              ;L'ARGOMENTO SUCCESSIVO NELLA LISTA
0283   C36902   R  542          JP PR0                ;LA LISTA CONTINUA
0286   CD7507   R  543  PR6     CALL CRLF             ;LA LISTA FINISCE
0289   CD8E00   R  544          CALL FINISH
028C   CD1D04   R  545  PR8     CALL EXPR             ;VALUTA L'ESPRESSIONE
028F   C5          546          PUSH BC
0290   CDBF06   R  547          CALL PRTNUM           ;STAMPA IL VALORE
0293   C1          548          POP BC
0294   C37B02   R  549          JP PR3                ;ALTRO DA STAMPARE?

              TINY.BASIC      21$GIUGNO$1979      PAGE   11
LOC    OBJ CODE M STMT SOURCE STATEMENT               BAU 4.1

                   550  *E
                   551  ;                                                                        ;
                   552  ;                                                                        ;
                   553  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   554  ;                                                                        ;
                   555  ;                                                                        ;
                   556  ;....................GOSUB.................RETURN..............................;
                   557  ;                                                                        ;
                   558  ;                                                                        ;
                   559  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   560  ;                                                                        ;
                   561  ;                                                                        ;
                   562  ; 'GOSUB <EXPR>;' O 'GOSUB <EXPR><CR>' SI COMPORTA COME IL COMANDO 'GOTO', ;
                   563  ; ECCETTO CHE IL PUNTATORE CORRENTE AL TESTO (CIOE' AL PROGRAMMA D'UTENTE), LO ;
                   564  ; STACK POINTER ECC. SONO SALVATI IN MODO DA POTER CONTINUARE L'ESECUZIONE DEL ;
                   565  ; PROGRAMMA DOPO LA 'RETURN' DA SUBROUTINE. PER OTTENERE L'ANNIDABILITA' DELLE ;
                   566  ; SUBROUTINE (ED ANCHE LA RECURSIVITA'), L'AREA DI SALVATAGGIO DEVE AVERE UNA ;
                   567  ; STRUTTURA A CATASTA. LO STACK POINTER VIENE SALVATO IN 'STKGOS',IL PRECEDENTE;
                   568  ; CONTENUTO DI 'STKGOS' VIENE SALVATO NELLO STACK.SE SI E' NEL PROGRAMMA PRIN- ;
                   569  ; CIPALE, 'STKGOS' CONTIENE ZERO (INIZIALIZZATO NELLA ROUTINE MAIN DELL'INTER- ;
                   570  ; PRETATORE),MA VIENE SALVATO ANCHE QUESTO PERCHE' SERVE AD INDICARE CHE NON ;
                   571  ; CI DEVONO ESSERE ULTERIORI 'RETURN'.                                   ;
                   572  ;                                                                        ;
                   573  ; 'RETURN<CR>' COMPIE A ROVESCIO TUTTE LE OPERAZIONI FATTE DA 'GOSUB' IN MODO ;
                   574  ; DA TORNARE AD INTERPRETARE IL COMANDO SUCCESSIVO ALLA 'GOSUB' INCONTRATA ;
                   575  ; PER ULTIMA. SE 'STKGOS' E' ZERO, SIGNIFICA CHE C'E' UNA 'RETURN' SENZA CHE ;
                   576  ; CI SIA PRIMA UNA 'GOSUB' ; SI E' QUINDI IN CONDIZIONE DI ERRORE.       ;
                   577  ;                                                                        ;
                   578  ;                                                                        ;
0297   CD4C07   R  579  GOSUB   CALL PUSHA            ;SALVA LE VARIABILI
029A   CD1D04   R  580          CALL EXPR             ;CORRENTI
029D   D5          581          PUSH DE               ;E IL PUNTATORE AL TESTO
029E   CD5706   R  582          CALL FNDLN            ;CERCA LA LINEA INDICATA
02A1   C20801   R  583          JP NZ AHOW            ;NON C'E':STAMPA 'HOW?'
02A4   2A0108   R  584          LD HL,(CURRNT)        ;TROVATA: SALVA I PRECEDENTI
02A7   E5          585          PUSH HL               ;'CURRNT' E 'STKGOS'
02A8   2A0308   R  586          LD HL,(STKGOS)
02AB   E5          587          PUSH HL
02AC   210000      588          LD HL,0               ;E CARICA QUELLI NUOVI
02AF   220708   R  589          LD (LOPVAR),HL
02B2   39          590          ADD HL,SP
02B3   220308   R  591          LD (STKGOS),HL
02B6   C31802   R  592          JP RUNTSL             ;ESEGUE LA LINEA DATA
02B9   CDD205   R  593  RETURN  CALL ENDCHK           ;CI DEVE ESSERE UN 'CR'
02BC   2A0308   R  594          LD HL,(STKGOS)        ;RECUPERA IL PRECEDENTE
02BF   7C          595          LD A,H                ;STACK POINTER: SE E' 0
02C0   B5          596          OR L                  ;VUOL DIRE CHE NON ESISTE
02C1   CAD805   R  597          JP Z,QWHAT            ;C'E' UN 'RET' DI TROPPO
02C4   F9          598          LD SP,HL              ;ALTRIMENTI LO RIPRISTINA
02C5   E1          599          POP HL                ;RIPRISTINA PURE IL
02C6   220308   R  600          LD (STKGOS),HL        ;PRECEDENTE 'STKGOS',
02C9   E1          601          POP HL
02CA   220108   R  602          LD (CURRNT),HL        ;IL PRECEDENTE 'CURRNT',
02CD   D1          603          POP DE                ;IL PUNTATORE AL TESTO,
02CF   CD3007   R  604          CALL POPA             ;E I PARAMETRI PRECEDENTI
02D1   CD8E00   R  605          CALL FINISH           ;CONTINUA AD ESEGUIRE IL PROGR.
```

```
                  433  ;
                  434  ; CI SONO TRE PUNTI DI ENTRATA IN 'RUN' :                                      ;
                  435  ; 'RUNNXL' CERCA LA LINEA SUCCESSIVA, MEMORIZZA IL SUO INDIRIZZO E LA ESEGUE   ;
                  436  ; 'RUNTSL' MEMORIZZA L'INDIRIZZO DELLA LINEA CORRENTE E LA ESEGUE              ;
                  437  ; 'RUNSML' CONTINUA L'ESECUZIONE SULLA STESSA LINEA                            ;
                  438  ;                                                                              ;
                  439  ; 'GOTO <EXPR><CR>' VALUTA L'ESPRESSIONE,CERCA LA LINEA D'ARRIVO E VA A RUNTSL ;
                  440  ;                                                                              ;
                  441  ;                                                                              ;
01FA  CDD205    R 442  NEW      CALL ENDCHK          ;ROUTINE ESECUZIONE COMANDO NEW
01FD  211508    R 443           LD HL,TXTBGN
0200  221308    R 444           LD (TXTUNF),HL
                  445  ;
0203  CDD205    R 446  STOP     CALL ENDCHK          ;ROUTINE ESECUZIONE COMANDO STOP
0206  C30000    R 447           JP START
                  448  ;
0209  CDD205    R 449  RUN      CALL ENDCHK          ;ROUTINE ESECUZIONE COMANDO RUN
020C  111508    R 450           LD DE,TXTBGN         ;PRIMA LINEA DI PROGRAMMA
                  451  ;
020F  210000      452  RUNNXL   LD HL,0
0212  CD5F06    R 453           CALL FNDLNP          ;CERCA LA LINEA CON NUMERO DATO
0215  DA0000    R 454           JP C,START           ;SE CY=1 HA SUPERATO LA FINE
                  455                                ;E ABBANDONA
0218  EB          456  RUNTSL   EX DE,HL             ;FA PUNTARE 'CURRNT' ALLA LINEA DATA

                       TINY.BASIC     21$GIUGNO$1979      PAGE   9
 LOC  OBJ CODE M STMT SOURCE STATEMENT                     BAU 4.1

0219  220108    R 457           LD (CURRNT),HL
021C  EB          458           EX DE,HL
021D  13          459           INC DE               ;SALTA IL NUMERO DELLA LINEA
021E  13          460           INC DE
                  461  ;
021F  CD9707    R 462  RUNSML   CALL CHKIO
0222  213301    R 463           LD HL,TAB2-1         ;CERCA IN TAB2 IL COMANDO
0225  C3CA01    R 464           JP EXEC              ;E LO ESEGUE
                  465  ;
0228  CD1D04    R 466  GOTO     CALL EXPR            ;VALUTA L'ESPR. CHE SEGUE 'GOTO'
022B  D5          467           PUSH DE              ;SALVA IL PUNTATORE PER
                  468                                ;LA ROUTINE DI ERRORE
022C  CDD205    R 469           CALL ENDCHK          ;DEVE TROVARE UN CARRIAGE-RETURN
022F  CD5706    R 470           CALL FNDLN           ;CERCA LA LINEA A CUI DEVE SALTARE
0232  C20801    R 471           JP NZ,AHOW           ;CHIEDE 'HOW?' SE NON
                  472                                ;HA TROVATO LA LINEA COL NUMERO DATO
0235  F1          473           POP AF               ;BILANCIA LO STACK
0236  C31802    R 474           JP RUNTSL            ;VA AD ESEGUIRE LA NUOVA LINEA
                  475  ;                                                                              ;
                  476  ;                                                                              ;
                  477  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  478  ;                                                                              ;
                  479  ;                                                                              ;
                  480  ;.......................LIST..................PRINT............................;
                  481  ;                                                                              ;
                  482  ;                                                                              ;
                  483  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  484  ;                                                                              ;
                  485  ;                                                                              ;
                  486  ; IL COMANDO 'LIST' HA DUE FORMATI :                                           ;
                  487  ; 'LIST<CR>' CHE STAMPA TUTTE LE LINEE DI PROGRAMMA MEMORIZZATE                ;
                  488  ; 'LIST <NUM><CR>' CHE STAMPA TUTTE LE LINEE DA QUELLA CON NUMERO <NUM>        ;
                  489  ; IL COMANDO PUO' ESSERE INTERROTTO BATTENDO CONTROL-C                         ;
                  490  ;                                                                              ;
                  491  ; IL COMANDO PRINT PUO' ESSERE 'PRINT <LISTA>;' OPPURE 'PRINT <LISTA><CR>'     ;
                  492  ; DOVE <LISTA> E' UNA LISTA DI ESPRESSIONI, FORMATI, BACK-ARROW E STRINGHE.    ;
                  493  ; TUTTI QUESTI TERMINI SONO SEPARATI DA VIRGOLE.                               ;
                  494  ;                                                                              ;
                  495  ; UN FORMATO E' UN SEGNO '#' SEGUITO DA UNA <EXPR> E INDICA QUANTE POSIZIONI   ;
                  496  ; DI STAMPA DEVONO ESSERE RISERVATE AL VALORE DELL'ESPRESSIONE DA STAMPARE.    ;
                  497  ; QUESTO VALORE RIMANE VALIDO FINCHE' NON SI TROVA UN ALTRO FORMAT.            ;
                  498  ; SE NON VIENE DATO ALCUN FORMAT VIENE PRESO 6 COME VALORE DI DEFAULT.         ;
                  499  ;                                                                              ;
                  500  ; UNA STRINGA DEVE ESSERE RACCHIUSA FRA APICI (') O FRA VIRGOLETTE (")          ;
                  501  ;                                                                              ;
                  502  ; UN BACK-ARROW (FRECCIA ALL'INDIETRO) FA UN RITORNO CARRELLO SENZA FAR AVAN-  ;
                  503  ; ZARE LA CARTA, CIOE' FA SCRIVERE SULLA STESSA LINEA DA CAPO                  ;
                  504  ;                                                                              ;
                  505  ; QUANDO E' FINITA TUTTA LA LISTA VIENE STAMPATO UN CR-LF. LO STESSO CAPITA    ;
                  506  ; SE NON C'E' LA LISTA DOPO 'PRINT'.                                           ;
                  507  ; SE INVECE LA LISTA TERMINA CON UNA VIRGOLA NON VIENE STAMPATO IL CR-LF       ;
                  508  ;                                                                              ;
                  509  ;                                                                              ;
0239  CDDD00    R 510  LIST     CALL TSTNUM          ;PROVA SE C'E' UN NUMERO
023C  CDD205    R 511           CALL ENDCHK          ;SE NON C'E' SI TROVA IL VALORE 0
023F  CD5706    R 512           CALL FNDLN           ;TROVA LA LINEA CORRENTE O LA SUCC.
0242  DA0000    R 513  LS1      JP C,START           ;C: HA SUPERATO LA FINE DEL PROGR.
0245  CD0107    R 514           CALL PRTLN           ;STAMPA LA LINEA

                       TINY.BASIC     21$GIUGNO$1979      PAGE  10
 LOC  OBJ CODE M STMT SOURCE STATEMENT                     BAU 4.1

0248  CD9707    R 515           CALL CHKIO           ;SI FERMA SE ARRIVA CONTROL-C
024B  CD5F06    R 516           CALL FNDLNP          ;CERCA LA LINEA SUCCESIVA
024E  C34202    R 517           JP LS1               ;E RIMANE IN LOOP
```

```
                       TINY.BASIC     21$GIUGNO$1979      PAGE  12
 LOC  OBJ CODE M STMT SOURCE STATEMENT                     BAU 4.1

                  606  *E
                  607  ;                                                                              ;
                  608  ;                                                                              ;
                  609  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  610  ;                                                                              ;
                  611  ;                                                                              ;
                  612  ;......................FOR...................NEXT..............................;
                  613  ;                                                                              ;
                  614  ;                                                                              ;
                  615  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  616  ;                                                                              ;
                  617  ;                                                                              ;
                  618  ; 'FOR' HA DUE FORMATI 'FOR <VAR>=<EXPR1> TO <EXPR2> STEP <EXPR3>'  OPPURE    ;
                  619  ; 'FOR <VAR>=<EXPR1> TO <EXPR2>'.LA SECONDA FORMA E' ANALOGA ALLA PRIMA ECCET- ;
                  620  ; TO CHE ASSUME AUTOMATICAMENTE PER EXPR3 IL VALORE +1 (CIOE' STEP=+1)         ;
                  621  ; L'INTERPRETATORE PONE LA VARIABILE VAR AL VALORE INDICATO DA EXPR1, CALCOLA ;
                  622  ; IL VALORE DI EXPR2 ED EXPR3 E LI SALVA, INSIEME CON IL PUNTATORE AL TESTO    ;
                  623  ; ECC., NELL'AREA DI SALVATAGGIO PER 'FOR', COSTITUITA DA 'LOPVAR', 'LOPINC', ;
                  624  ; 'LOPLMT', 'LOPLN', E 'LOPPT'.SE VI E' GIA' QUALCOSA NELL'AREA DI SALVATAGGIO ;
                  625  ; (INDICATO DA 'LOPVAR' DIVERSO DA ZERO), PRIMA DI SCRIVERE I NUOVI DATI NELLA ;
                  626  ; SAVEAREA, QUELLI GIA' PRESENTI VENGONO SALVATI NELLO STACK.                  ;
                  627  ; L'INTERPRETATORE SCAVA NELLO STACK PER CERCARE SE ERA GIA' STATA USATA LA    ;
                  628  ; STESSA VARIABILE IN UN ALTRO LOOP NON ANCORA CHIUSO. SE CAPITA QUESTO ,IL    ;
                  629  ; LOOP PRECEDENTE VIENE ELIMINATO.                                             ;
                  630  ;                                                                              ;
                  631  ; 'NEXT <VAR>' FUNZIONA COME FINE LOGICA (NON NECESSARIAMENTE FISICA) DEL LOOP.;
                  632  ; LA VARIABILE DI CONTROLLO VIENE CONFRONTATA CON LOPVAR. SE NON E' LA STESSA  ;
                  633  ; L'INTERPRETATORE CERCA NELLO STACK QUELLA ESATTA ED ELIMINA TUTTE LE ALTRE   ;
                  634  ; CHE NON CORRISPONDONO. IN ENTRAMBI I CASI IL VALORE DELLA VARIABILE VIENE    ;
                  635  ; INCREMENTATO DI 'STEP' ED IL RISULTATO E' CONFRONTATO CON IL VALORE LIMITE   ;
                  636  ; SE QUESTO NON E' ANCORA SUPERATO , IL CONTROLLO TORNA ALL'ISTRUZIONE SUCCES- ;
                  637  ; SIVA A QUELLA DI FOR. SE INVECE IL VALORE LIMITE E' SUPERATO, VIENE PULITA   ;
                  638  ; LA SAVEAREA  E IL PROGRAMMA PROSEGUE AD INTERPRETARE LE ISTRUZIONI SUCCESSIVE;
                  639  ;                                                                              ;
                  640  ;                                                                              ;
02D4  CD4C07    R 641  FOR      CALL PUSHA           ;SALVA LA SAVEAREA PRECEDENTE
02D7  CDA605    R 642           CALL SETVAL          ;INIZIALIZZA LA VARIABILE DEL LOOP
02DA  2B          643           DEC HL               ;PUNTATA DA HL
02DB  220708    R 644           LD (LOPVAR),HL       ;SALVA IL PUNTATORE ALLA VAR.
02DE  21A201    R 645           LD HL,TAB5-1         ;USA 'EXEC' PER CERCARE IL 'TO'
02E1  C3CA01    R 646           JP EXEC
02E4  CD1D04    R 647  FR1      CALL EXPR            ;VALUTA IL VALORE FINALE
02E7  220B08    R 648           LD (LOPLMT),HL       ;E LO SALVA
02EA  21A801    R 649           LD HL,TAB6-1         ;USA 'EXEC' PER CERCARE LO 'STEP'
02ED  C3CA01    R 650           JP EXEC
02F0  CD1D04    R 651  FR2      CALL EXPR            ;TROVATO, CALCOLA IL PASSO
02F3  C3F902    R 652           JP FR4
02F6  210100      653  FR3      LD HL,1              ;NON C'E' LO STEP,LO METTE A 1
02F9  220908    R 654  FR4      LD (LOPINC),HL       ;SALVA ANCHE IL PASSO,
02FC  2A0108    R 655  FR5      LD HL,(CURRNT)       ;IL NUMERO DELLA LINEA CORRENTE
02FF  220D08    R 656           LD (LOPLN),HL
0302  EB          657           EX DE,HL             ;E IL PUNTATORE AL TESTO
0303  220F08    R 658           LD (LOPPT),HL
0306  010A00      659           LD BC,0AH            ;SCAVA NELLO STACK
0309  2A0708    R 660           LD HL,(LOPVAR)       ;PER TROVARE 'LOPVAR'
030C  EB          661           EX DE,HL
030D  60          662           LD H,B
030E  68          663           LD L,B               ;AZZERA HL

                       TINY.BASIC     21$GIUGNO$1979      PAGE  13
 LOC  OBJ CODE M STMT SOURCE STATEMENT                     BAU 4.1

030F  39          664           ADD HL,SP            ;METTE LO STACK POINTER IN HL
0310  3E          665           DEFB 3EH             ;ISTRUZIONE DI SKIP
0311  09          666  FR7      ADD HL,BC            ;OGNI LIVELLO E' PROFONDO 10 BYTE
0312  7E          667           LD A,(HL)            ;CERCA LA PRECEDENTE LOPVAR
0313  23          668           INC HL
0314  B6          669           OR (HL)
0315  CA3203    R 670           JP Z,FR8             ;0 SIGNIFICA CHE NON CE N'E' PIU'
0318  7E          671           LD A,(HL)
0319  2B          672           DEC HL
031A  BA          673           CP D                 ;E' LA STESSA ?
031B  C21103    R 674           JP NZ,FR7
031E  7E          675           LD A,(HL)
031F  BB          676           CP E                 ;ANCHE L'ALTRA META' E' UGUALE?
0320  C21103    R 677           JP NZ,FR7
0323  EB          678           EX DE,HL             ;TROVATA
0324  210000      679           LD HL,0
0327  39          680           ADD HL,SP            ;AGGIORNA LO STACK POINTER
0328  44          681           LD B,H
0329  4D          682           LD C,L
032A  210A00      683           LD HL,0AH
032D  19          684           ADD HL,DE
032E  CD2107    R 685           CALL MVDOWN          ;ELIMINA 10 BYTE DALLO STACK
0331  F9          686           LD SP,HL
0332  2A0F08    R 687  FR8      LD HL,(LOPPT)        ;LAVORO ESEGUITO,RIPRISTINA DE
0335  EB          688           EX DE,HL
0336  CD8E00    R 689           CALL FINISH          ;CONTINUA
                  690  ;
0339  CD9500    R 691  NEXT     CALL TSTV            ;CHIEDE L'IND. DELLA VAR. DI LOOP
```

```
033C   DAD805   R  692            JP C,QWHAT           ;NON C'E' LA VAR. : STAMPA 'WHAT?'
033F   220508   R  693            LD (VARNXT),HL       ;C'E' LA VAR. NE SALVA L'IND.
0342   D5          694   NX0      PUSH DE              ;SALVA IL PUNTATORE AL TESTO
0343   EB          695            EX DE,HL
0344   2A0708   R  696            LD HL,(LOPVAR)       ;CARICA LA VAR. IN 'FOR'
0347   7C          697            LD A,H               ;E' 0?
0348   B5          698            OR L                 ;SE SI' C'E' UN 'NEXT' DI TROPPO
0349   CAD905   R  699            JP Z,AWHAT           ;E SEGNALA L'ERRORE
034C   CD8000   R  700            CALL COMP            ;LA VAR. DELL'ULTIMO 'FOR'
034F   CA5C03   R  701            JP Z,NX3             ;E' LA STESSA DI QUESTO NEXT?
0352   D1          702            POP DE               ;NO,ELIMINA IL 'LOOP' CORRENTE
0353   CD3007   R  703            CALL POPA            ;TOGLIE UN LIVELLO DALLO STACK
0356   2A0508   R  704            LD HL,(VARNXT)       ;E CERCA ANCORA
0359   C34203   R  705            JP NX0
035C   5E          706   NX3      LD E,(HL)            ;SI',LA VARIABILE DEL NEXT
035D   23          707            INC HL               ;E' LA STESSA DELL'ULTIMO FOR
035E   56          708            LD D,(HL)            ;DE CONTIENE IL VALORE DELLA VAR.
035F   2A0908   R  709            LD HL,(LOPINC)
0362   E5          710            PUSH HL
0363   19          711            ADD HL,DE            ;INCREMENTA LA VAR. DI UN PASSO
0364   EB          712            EX DE,HL
0365   2A0708   R  713            LD HL,(LOPVAR)       ;SALVA IL VALORE AGGIORNATO
0368   73          714            LD (HL),E
0369   23          715            INC HL
036A   72          716            LD (HL),D
036B   2A0B08   R  717            LD HL,(LOPLMT)       ;HL -> VALORE FINALE
036E   F1          718            POP AF               ;RECUPERA IL PRECEDENTE HL
036F   B7          719            OR A                 ;POSIZIONA I FLAG
0370   F27403   R  720            JP P,NX1             ;STEP > 0
0373   EB          721            EX DE,HL             ;STEP < 0

                          TINY.BASIC      21$GIUGNO$1979       PAGE  14
LOC    OBJ CODE M STMT SOURCE STATEMENT                         BAU 4.1

0374   CD9C05   R  722   NX1      CALL CKHLDE          ;CONFRONTA COL VALORE FINALE
0377   D1          723            POP DE               ;RECUPERA IL PUNTATORE AL TESTO
0378   DA8803   R  724            JP C,NX2             ;OLTRE IL VALORE FINALE
037B   2A0D08   R  725            LD HL,(LOPLN)        ;ENTRO IL VALORE FINALE
037E   220108   R  726            LD (CURRNT),HL       ;TORNA A 'CURRNT' E AL
0381   2A0F08   R  727            LD HL,(LOPPT)        ;PUNTATORE AL TESTO
0384   EB          728            EX DE,HL             ;SALVATI PRIMA
0385   CD8E00   R  729            CALL FINISH          ;CONTINUA
0388   CD3007   R  730   NX2      CALL POPA            ;ELIMINA QUESTO LOOP
038B   CD8E00   R  731            CALL FINISH
                   732   ;                                                                      ;
                   733   ;                                                                      ;
                   734   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   735   ;                                                                      ;
                   736   ;                                                                      ;
                   737   ;.............REM.........IF..........INPUT..........LET.(DEFAULT).............;
                   738   ;                                                                      ;
                   739   ;                                                                      ;
                   740   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   741   ;                                                                      ;
                   742   ;                                                                      ;
                   743   ; 'REM' PUO' ESSERE SEGUITO DA QUALUNQUE COSA CHE VIENE IGNORATA DALL'INTERPRE-;
                   744   ; TATORE. 'REM' VIENE CONSIDERATO COME UN 'IF' CON CONDIZIONE NON VERIFICATA   ;
                   745   ;                                                                      ;
                   746   ; 'IF' E' SEGUITO DA UNA ESPRESSIONE CHE FA DA CONDIZIONE E DA UNO O PIU' CO-  ;
                   747   ; MANDI (INCLUSI ALTRI 'IF') SEPARATI DA ';'.NOTARE CHE LA PAROLA 'THEN' NON   ;
                   748   ; E' USATA. L'INTERPRETATORE VALUTA L'ESPRESSIONE, E SE E' DIVERSA DA ZERO,    ;
                   749   ; PROSEGUE L'INTERPRETAZIONE. SE INVECE L'ESPRESSIONE VALE ZERO, I COMANDI     ;
                   750   ; SUCCESSIVI VENGONO IGNORATI E L'ESECUZIONE RIPRENDE DALLA LINEA SUCCESSIVA.  ;
                   751   ;                                                                      ;
                   752   ; 'INPUT' SI COMPORTA QUASI COME IL COMANDO 'PRINT' ED E' SEGUITO DA UNA LISTA ;
                   753   ; DI VOCI. SE UNA DI QUESTE VOCI E' UNA STRINGA FRA APICI O VIRGOLETTE, O E'   ;
                   754   ; UN BACK-ARROW, QUESTA SI COMPORTA ESATTAMENTE COME IN 'PRINT'. SE UNA VOCE   ;
                   755   ; E' UNA VARIABILE,NE VIENE STAMPATO IL NOME SEGUITO DA ':', POI L'INTERPRETA- ;
                   756   ; TORE ASPETTA IN INGRESSO UNA ESPRESSIONE. ASSEGNA QUINDI IL VALORE DELL'     ;
                   757   ; ESPRESSIONE ALLA VARIABILE. SE LA VARIABILE E' PRECEDUTA DA UNA STRINGA,     ;
                   758   ; VIENE STAMPATA LA STRINGA SEGUITA DAI ':' E SENZA STAMPARE IL NOME DELLA     ;
                   759   ; VARIABILE. L'INTERPRETATORE SI COMPORTA POI COME PRECEDENTEMENTE DETTO       ;
                   760   ;                                                                      ;
                   761   ; SE L'ESPRESSIONE BATTUTA DA TASTIERA NON E' VALIDA, L'INTERPRETATORE STAMPA  ;
                   762   ; 'WHAT?' OPPURE 'HOW?' OPPURE 'SORRY',RIPRESENTA LA DOMANDA E ATTENDE L'      ;
                   763   ; ESPRESSIONE GIUSTA. QUESTO COMPORTAMENTO CONTINUA FINCHE' NON VIENE BATTUTO  ;
                   764   ; UN CONTROL-C : LA ROUTINE DI GESTIONE PER QUESTO CASO E' 'INPERR'.           ;
                   765   ;                                                                      ;
                   766   ; 'LET' E' SEGUITO DA UNA LISTA DI VOCI SEPARATE DA VIRGOLE. OGNI VOCE CONSI-  ;
                   767   ; STE DI UNA VARIABILE, UN SEGNO DI UGUALE E UNA ESPRESSIONE. L'INTERPRETATORE ;
                   768   ; VALUTA L'ESPRESSIONE E NE ASSEGNA IL VALORE ALLA VARIABILE. ANCHE SE NON     ;
                   769   ; C'E' LA PAROLA 'LET', L'INTERPRETATORE ESEGUE LE STESSE OPERAZIONI. QUESTO   ;
                   770   ; CASO E' TRATTATO DALLA ROUTINE 'DEFLT'                                       ;
                   771   ;                                                                      ;
                   772   ;                                                                      ;
038E   210000      773   REM      LD HL,0              ;REM E' COME UN IF SEMPRE FALSO
0391   C39703   R  774            JP IF1               ;AZZERA LA VAR. LOGICA E VA AD IF
                   775   ;
0394   CD1D04   R  776   IF       CALL EXPR            ;VALUTA L'ESPR. DOPO IF
0397   7C          777   IF1      LD A,H               ;VALE 0?
0398   B5          778            OR L
0399   C21F02   R  779            JP NZ,RUNSML         ;NO, CONTINUA AD ESEGUIRE LA LINEA

                          TINY.BASIC      21$GIUGNO$1979       PAGE  15
LOC    OBJ CODE M STMT SOURCE STATEMENT                         BAU 4.1

039C   CD7706   R  780            CALL FNDSKP          ;SI',SALTA IL RESTO DELLA LINEA
039F   D21802   R  781            JP NC,RUNTSL         ;E PASSA AD ESEGUIRE LA SUCCESSIVA
03A2   C30000   R  782            JP START             ;SE NON C'E' LA LINEA SUCCESSIVA
                   783   ;                             ;SI RIPRESENTA
03A5   2A0508   R  784   INPERR   LD HL,(STKINP)       ;GESTIONE ERRORE SULL'INPUT
03A8   F9          785            LD SP,HL             ;RIPRISTINA LO STACK POINTER
03A9   E1          786            POP HL               ;E IL 'CURRNT' PRECEDENTI
03AA   220108   R  787            LD (CURRNT),HL
03AD   D1          788            POP DE               ;RIPRISTINA IL PRECEDENTE
03AE   D1          789            POP DE               ;PUNTATORE AL TESTO
                   790   ;
                   791   INPUT
03AF   D5          792   IP1      PUSH DE              ;SALVA PUNT. PER EVENTUALI ERRORI
03B0   CD8F06   R  793            CALL QTSTG           ;LA PROSSIMA VOCE E' UNA STRINGA?
03B3   C3BF03   R  794            JP IP2               ;NO
03B6   CD9500   R  795            CALL TSTV            ;SI'.E' SEGUITA DA UNA VARIABILE?
03B9   DAFD03   R  796            JP C,IP4             ;NO
03BC   C3D103   R  797            JP IP3               ;SI':FA ENTRARE LA VARIABILE
03BF   D5          798   IP2      PUSH DE              ;SALVA PER 'PRTSTG'
03C0   CD9500   R  799            CALL TSTV            ;CI DEVE ESSERE UNA VARIABILE ORA
03C3   DAD805   R  800            JP C,QWHAT           ;SE NON C'E',SEGNALA L'ERRORE
03C6   1A          801            LD A,(DE)            ;SI PREPARA PER 'PRTSTG'
03C7   4F          802            LD C,A
03C8   97          803            SUB A
03C9   12          804            LD (DE),A
03CA   D1          805            POP DE
03CB   CD8106   R  806            CALL PRTSTG          ;STAMPA LA STRINGA
03CE   79          807            LD A,C               ;RIPRISTINA IL TESTO
03CF   1B          808            DEC DE
03D0   12          809            LD (DE),A
03D1   D5          810   IP3      PUSH DE              ;SALVA PUNT. PER EVENTUALI ERRORI
03D2   EB          811            EX DE,HL
03D3   2A0108   R  812            LD HL,(CURRNT)       ;SALVA ANCHE 'CURRNT'
03D6   E5          813            PUSH HL
03D7   21AF03   R  814            LD HL,IP1            ;CURRNT -> A UN NUMERO NEGATIVO
03DA   220108   R  815            LD (CURRNT),HL       ;CIOE' AL COD. OP. DI 'PUSH DE'
03DD   210000      816            LD HL,0              ;SALVA ANCHE LO STACK POINTER
03E0   39          817            ADD HL,SP
03E1   220508   R  818            LD (STKINP),HL
03E4   D5          819            PUSH DE              ;IN DE C'E' IL VECCHIO HL
03E5   3E3A        820            LD     A,':'         ;STAMPA IL ':'
03E7   CD0E06   R  821            CALL GETLN           ;RICEVE UNA LINEA
03EA   11371F   @  822            LD DE,BUFFER         ;PUNTA AL BUFFER
03ED   CD1D04   R  823            CALL EXPR            ;VALUTA L'ESPRESSIONE IN INGRESSO
03F0   00          824            NOP                  ;SI PUO' METTERE UNA 'CALL ENDCHK'
03F1   00          825            NOP                  ;PER VERIFICARE CHE L'ESPRESSIONE
03F2   00          826            NOP                  ;IN INGRESSO TERMINI CON 'CR'
03F3   D1          827            POP DE               ;OK, RECUPERA IL VECCHIO 'CURRNT'
03F4   EB          828            EX DE,HL             ;SALVA IL VALORE IN VAR.
03F5   73          829            LD (HL),E
03F6   23          830            INC HL
03F7   72          831            LD (HL),D
03F8   E1          832            POP HL               ;RECUPERA IL 'CURRNT' PRECEDENTE
03F9   220108   R  833            LD (CURRNT),HL
03FC   D1          834            POP DE               ;E IL PUNTATORE AL TESTO
03FD   F1          835   IP4      POP AF               ;BILANCIA LO STACK
03FE   CDC900   R  836            CALL TSTC            ;IL PROSSIMO CARATTERE E' ',' ?
0401   2C          837            DEFB ','

                          TINY.BASIC      21$GIUGNO$1979       PAGE  16
LOC    OBJ CODE M STMT SOURCE STATEMENT                         BAU 4.1

0402   03          838            DEFB (IP5-$-1)&0FFH
0403   C3AF03   R  839            JP IP1               ;SI',CI SONO ALTRI TERMINI
0406   CD8E00   R  840   IP5      CALL FINISH          ;CONTINUA
                   841   ;
0409   1A          842   DEFLT    LD A,(DE)
040A   FE0D        843            CP CR                ;UNA LINEA VUOTA E' ACCETTABILE
040C   CA1A04   R  844            JP Z,LT1             ;SE NO E' UNA 'LET'
                   845   ;
040F   CDA605   R  846   LET      CALL SETVAL          ;ASSEGNA IL VALORE ALLA VARIABILE
0412   CDC900   R  847            CALL TSTC
0415   2C          848            DEFB ','             ;SE C'E' ',' CI SONO ALTRE
0416   03          849            DEFB (LT1-$-1)&0FFH  ;ASSEGNAZIONI DA FARE
0417   C30F04   R  850            JP LET
041A   CD8E00   R  851   LT1      CALL FINISH          ;CONTINUA
```

ECCO IL PERSONAL COMPUTER DEI VOSTRI SOGNI

- Personal computer integrato con doppia unità a disco
- Unità centrale Z-80
- Estesa dotazione di software
- Monitor professionale a fosforo verde o giallo
- Tastiera 76 tasti con keypad numerico separato di altissima qualità
- Costruzione interamente metallica
- Uscita audio con amplificatore ed altoparlante incorporato
- Interfacce parallele e seriali 20 mA ed RS-232
- Possibilità di usare un registratore come periferica di massa
- Dotazione standard 16K di memoria RAM espandibile a 48K
- Monitor/Debugger su epROM
- Uscita per stampante termica o professionale
- Prezzi a partire da 1.731.000.= lire

Finalmente non vi dovrete più accntentare dell'approssimativo, del computer giocattolo ove ogni finitura è sacrificata sull'altare dalla economia. Il nuovo Modello T della General Processor ha tutte le caratteristiche di un apparecchio di costo elevatissimo e costa poco di più di altri personal dalle caratteristiche nettamente inferiori.

Prima di decidere un acquisto esaminate da vicino le finiture di altissimo livello del Modello T, la sua costruzione robusta, la nitidezza del suo display professionale a fosforo verde, l'ergonomicità della sua tastiera e confrontate attentamente tutte le sue caratteristiche. Per ricevere una descrizione dettagliata del modello T inviate 1000 lire in francobolli al nostro indirizzo. Qualunque sia il problema da risolvere, il Modello T lo fa nel modo migliore ed il costo minore. General Processor: tecnologia italiana e prodotti di classe.

MODELLO "T"

Concessionari:

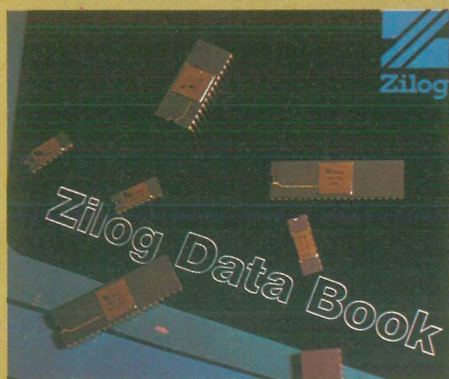
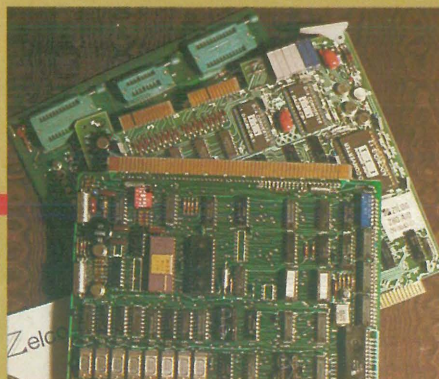
MILANO - 3G Elettronica - Via Conservatorio, 24 (tel. 79.34.71)
SANREMO - Unelco - Via Roma, 146 (tel. 88.38.92)
VENEZIA - CM Elettronica - Via Pegaso, 48 - Sottomarina di Chioggia

AREZZO - TECEM - Via IV Novembre
COSENZA - Studio Dr. Tripodi - Via Negressi - S. Giovanni in Fiore (tel. 99.21.42)

Si cercano concessionari per zone libere.

Via Montebello 3R - Firenze Tel. (055) 21.91.43

E' DURO PER ALCUNI DOVER AMMETTERE DI AVER PERSO IL "QUASI MONOPOLIO". I FATTI SONO QUELLI CHE CONTANO:

LA REALTÁ DEI MICRO COMINCIA DALLA

DUE FISIONOMIE DIVERSE MA UNICHE SOTTO MOLTI ASPETTI

Zilog

- E' l'unica casa completamente dedicata ai micro.
- Progetta tenendo presente i problemi globali ed in funzione di questi ottimizza i particolari perciò:
- vi offre di meglio come componenti, come piastre, come sistemi e come software.
- Ha alle spalle il più forte gruppo finanziario del mondo;
- ciò le consente di avere i migliori specialisti mondiali di hardware, di software e di sistemistica e da
- la sicurezza di disporre dei finanziamenti necessari a conservare la supremazia nel proprio campo (per questo Vi sono ditte anche in italia, che già nel 1979 hanno impostato la produzione di centinaia di migliaia di macchine con lo Z80).

Zelco

- E' esclusivamente dedicata ai micro e dà un supporto che va dalla fornitura di componenti, agli accessori (terminali, stampanti etc.) per sistemi
- E' composta da specialisti e non da venditori per offrire un servizio più rapido e non una assistenza che si ferma di fronte ai problemi nuovi;
- Usa essa stessa intensamente i calcolatori che vende, li conosce a fondo e produce software sofisticato per se e per gli altri.
- Ha una organizzazione agile che le permette di non ricaricare spese eccessive sui prodotti che vende e di rispondere in "real time". (Provare per credere).
- Ha una consegna pronta; ciò significa che non dovete aspettare due mesi per un SIO o una CPU e neppure per un calcolatore.
- Vi assiste direttamente per riparazioni o consulenza qualificata.

La nostra pubblicità maggiore è fatta da tre elementi:

1 - I nostri clienti **2 - La nostra qualità** **3 - I nostri prezzi**

Fate i confronti: per noi basta telefonare al 02/803336/804247
(nelle zone di Roma al 06/8102836)

Zelco S.r.l. Via V. Monti 21, 20123 Milano

23 OTTOBRE

Si terrà a Milano il seminario Zilog
Per informazioni e prenotazioni rivolgersi alla **Zelco**

LA NOTA

Un nuovo mercato per i microcomputer

di **Aldo Cavalcoli**

A livello nazionale è in corso un interessante ed intenso dibattito sul problema dell'inserimento della microinformatica nelle scuole.

Il discorso è molto articolato, in quanto se da un lato le scuole secondarie, istituti tecnici, necessitano rapidamente di un aggiornamento dei programmi di studio di informatica ed elettronica per tener conto degli sviluppi tecnologici, dall'altro le industrie hanno bisogno di una forza lavoro qualificata, con basi hardware e software sui microprocessori.

Ancora, la scuola è vista dai costruttori e rivenditori come potenziale mercato, anche se si parla di percentuali dell'ordine del 20% sul mercato potenziale globale.

Si scontrano allora due tendenze contrastanti:

1) la scuola deve fornire tecnici preparati come l'industria desidera, quindi occorre supportare la scuola con attrezzature e metodologie nuove, ma a questo scopo definite

2) la scuola è un mercato, per cui occorre conquistarlo per piazzare prodotti e filosofie tecniche polarizzate, per una successiva espansione delle linee di date case costruttrici.

La scuola, a sua volta, deve fare i conti con una teorica linea di sviluppo e con la realtà delle cose:

1) la scuola deve essere indipendente e definire al suo interno metodologie e strumenti

2) la scuola deve venire in contatto con gli usuali strumenti professionali che gli studenti si troveranno a gestire nella loro attività professionale.

La mediazione pare, per il momento, centrata sulle cosiddette "apparecchiature didattiche", spesso proposte da non costruttori di componenti, quindi meno sospette.

Ma, in tal caso, l'apparecchiatura didattica può diventare anche solo un prodotto in sé, a parte le opportune distinzioni, non agganciato ad uno sviluppo verso l'alto, verso la configurazione industriale reale, negando, almeno in parte, il contatto con gli usuali metodi e problematiche di lavoro progettuale.

In sostanza, una soluzione può essere data solo da una presa di posizione critica del corpo docente che si troverà a gestire la materia, in quanto tutte le fonti di approvvigionamento di materiale e know-how hanno senz'altro qualcosa di utile da dare alla scuola, solo che questo va individuato e scelto dall'utente finale, cioè la scuola stessa, che con un ruolo attivo non solo chiede, ma impara la definizione di prodotti con caratteristiche precise, secondo le sue esigenze.

Si è detto "ruolo attivo", ma per avere un ruolo attivo occorre essere nella materia.

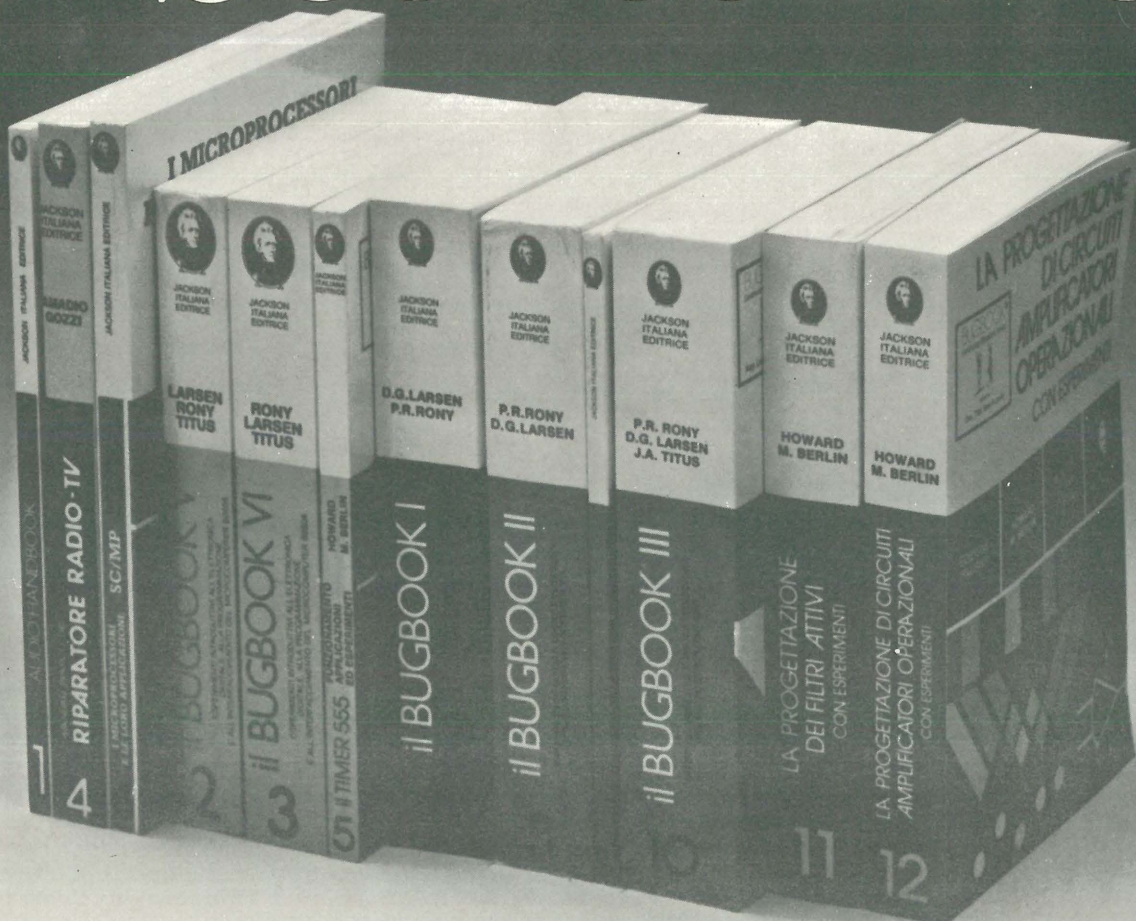
A questo punto si corre il rischio di tornare all'inizio, perché per entrare nella materia bisogna sperimentare e studiare, con il rischio di essere polarizzati verso certe linee di prodotti cui le case spesso allegano corsi di formazione gratuiti o quasi.

Anche qui nulla di male, perché più si conosce, più poi si può criticare.

Inizia anche a notarsi una certa tendenza autonoma sia di scuole che di singoli docenti, nel catturare esperienze e conoscenze di base per un successivo personale sviluppo dell'argomento.

Tutto questo indica che si è in movimento, che si è in una fase di saggio e sperimentazione.

Quello che occorre ora è un corretto e consapevole atteggiamento delle case costruttrici, che vedono le scuole non solo come un mercato, ma anche come lo strumento di formazione per i loro futuri tecnici.

i "best-sellers"

AUDIO HANDBOOK

Un manuale di progettazione audio con discussioni particolareggiate e progetti completi.

L. 9.500 (Abb. L. 8.550)

MANUALE PRATICO DEL RIPARATORE RADIO-TV

Un autentico strumento di lavoro. Fra i numerosi argomenti trattati figurano: il laboratorio. Il servizio a domicilio. Antenne singole e centralizzate. Riparazione dei TV b/n e colore. Il ricevitore AM FM. Apparecchi e BF e CB. Strumentazione. Elenco ditte di radiotecnica, ecc.

L. 18.500 (Abb. L. 16.650)

SC/MP

Questo testo sul microprocessore SC/MP è corredato da una serie di esempi di applicazione e di programmi di utilità generale, tali da permettere al lettore una immediata verifica dei concetti teorici esposti e un'immediata sperimentazione anche a livello di realizzazione progettuale.

L. 9.500 (Abb. L. 8.550)

IL BUGBOOK V E IL BUGBOOK VI

Esperimenti introduttivi all'elettronica digitale, alla programmazione ed all'interfacciamento del microprocessore 8080A. I Bugbook V e VI costituiscono i primi veri testi organici a livello universitario sui microprocessori, con taglio nettamente sperimentale. Questi testi, oltre al Virginia Polytechnic Institute, sono utilizzati in corsi aziendali, in seminari di aggiornamento tecnico e in scuole di tutto il mondo.

L. 19.000 ogni volume (Abb. L. 17.100)

IL TIMER 555

Il 555 è un temporizzatore dai mille usi. Il libro descrive circa 100 circuiti utilizzanti questo dispositivo e numerosi esperimenti.

L. 8.600 (Abb. L. 7.750)

IL BUGBOOK I E IL BUGBOOK II

Strumenti di studio per i neofiti e di aggiornamento professionale per chi già vive l'elettronica "tradizionale", questi due libri complementari presentano esperimenti sui circuiti logici e di memoria, utilizzanti circuiti integrati TTL. La teoria è subito collegata alla sperimentazione pratica, secondo il principio per cui si può veramente imparare solo quello che si sperimenta in prima persona.

L. 18.000 ogni volume (Abb. L. 16.200)

IL BUGBOOK II/A

Esperimenti di interfacciamento e trasmissione dati utilizzanti il ricevitore/trasmettitore universale asincrono (Uart) ed il Loop di corrente a 20 mA.

L. 4.500 (Abb. L. 4.050)

IL BUGBOOK III

Questo libro fornisce una parola definitiva sull'argomento "8080A" divenuto ormai un classico nella letteratura tecnica sui microprocessori. Da ogni parte , sia da istituti di formazione che da varie case costruttrici, sono stati pubblicati manuali e libri di testo, ma nessuno raggiunge la completezza di questo Bugbook e, soprattutto, nessuno presenta l'oggetto "8080A" in un modo così didattico e sperimentale.

L. 19.000 (Abb. L. 17.100)

LA PROGETTAZIONE DEI FILTRI ATTIVI CON ESPERIMENTI

Tratta un argomento di notevole attualità, rendendolo piano e comprensibile a tutti. Le riviste di settore dedicano ampio spazio a questo aspetto dell'elettronica da oltre tre anni. Questo libro raccoglie tutto quanto è necessario sapere sui filtri attivi aggiungendovi numerosi esempi pratici ed esperimenti.

L. 15.000 (Abb. L. 13.500)

LA PROGETTAZIONE DEGLI AMPLIFICATORI OPERAZIONALI CON ESPERIMENTI

Gli amplificatori operazionali, in gergo chiamati OP-AMP, sono ormai diffusissimi in elettronica. Il libro ne spiega il funzionamento illustra alcune applicazioni pratiche e forniscé numerosi esperimenti. Le persone interessate all'argomento sono moltissime: dal tecnico esperto al semplice hobbista. Si tratta del miglior libro pubblicato nella materia specifica.

L. 15.000 (Abb. L. 13.500)

Bit

CEDOLA DI COMMISSIONE LIBRARIA Da inviare a Jackson Italiana Editrice srl - Piazzale Massari, 22 - 20125 Milano.

SCONTO 10% AGLI ABBONATI

☐ Inviatemi i seguenti volumi pagherò al postino l'importo indicato più le spese di spedizione.

Nome ____________

Cognome ____________

Via ____________ N. ______

Città ____________ Cap. ______

Codice Fiscale ____________

Data ____________ Firma ____________

☐ Pagamento anticipato senza spese di spedizione.

N. —— Audio Handbook .... **L. 9.500** (Abb. L. 8.550)
N. —— Manuale del Riparatore Radio-TV .... **L. 18.500** (Abb. L. 16.650)
N. —— SC/MP .... **L. 9.500** (Abb. L. 8.550)
N. —— Bugbook V .... **L. 19.000** (Abb. L. 17.100)
N. —— Bugbook VI .... **L. 19.000** (Abb. L. 17.100)
N. —— Timer 555 .... **L. 8.600** (Abb. L. 7.750)
N. —— Bugbook I .... **L. 18.000** (Abb. L. 16.200)
N. —— Bugbook II .... **L. 18.000** (Abb. L. 16.200)
N. —— Bugbook II/A .... **L. 4.500** (Abb. L. 4.050)
N. —— Bugbook III .... **L. 19.000** (Abb. L. 17.100)
N. —— La Progettazione dei Filtri Attivi .... **L. 15.000** (Abb. L. 13.500)
N. —— La Progettazione degli Amp. Op. .... **L. 15.000** (Abb. L. 13.500)

Un gioco di logica: inversione

Tratto da "Basic Computer Games". Creative Computing Press*

Scopo del gioco è ordinare una lista di nove numeri (dall'1 al 9), in ordine casuale, nell'ordine naturale; per far questo si comanda al calcolatore di invertire l'ordine di un certo numero di numeri, calcolato a partire dalla sinistra della lista. Ad esempio, se la lista su cui operare è:

2 3 4 5 1 6 7 8 9

e si ordina di invertire 4 numeri, si otterrà:

5 4 3 2 1 6 7 8 9

Se a questo punto si invertono 5 numeri, si vince.

Si possono adottare strategie diverse, ma che in definitiva si riducono a due tipi di approccio: algoritmico ed euristico.

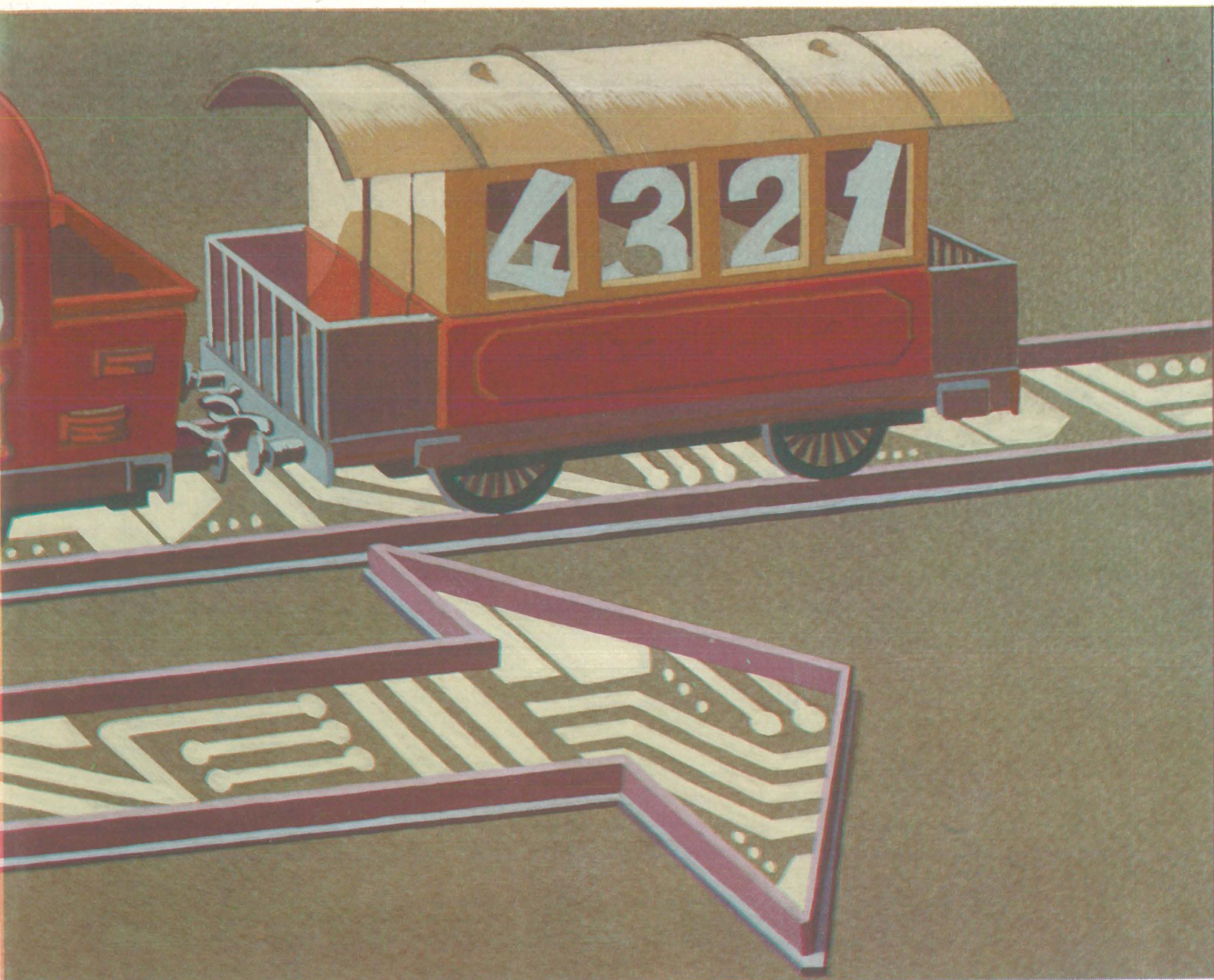
Il gioco offre l'interessante possibilità di cimentarsi con questi concetti in un contesto non teorico, ma pratico.

L'approccio algoritmico garantisce la soluzione in un numero di mosse prevedibile, essendo dato il numero di cifre che compongono la lista.

Ad esempio, seguendo un certo procedimento, si arriverà alla soluzione in 2N-3 mosse, posto che la lista sia costituita da N numeri.

L'elemento caratterizzante delle strategie di tipo

---

* Distribuito in Italia dalla Jackson Editrice s.r.l.

algoritmico è che permettono di sapere quale sarà la prossima mossa..

Questo tipo di metodo può essere facilmente implementato mediante il calcolatore.

L'approccio euristico si avvale di "ordinamenti parziali" all'interno della lista istante per istante.

Con questo tipo di approccio, la prossima mossa dipende dalla configurazione attuale della lista.

Non viene garantita la soluzione di un numero prevedibile di mosse, ma, con un po' di fortuna e di abilità, si può arrivare alla soluzione più rapidamente che non seguendo l'iter predeterminato del metodo algoritmico.

Questo tipo di metodo non è facilmente implementabile mediante il calcolatore.

In pratica, per lo più viene adottata una strategia "mista", con procedimenti in parte algoritmici, in parte euristici.

Ma va detto che, a parte la comodità di questo compromesso ai fini del gioco, vale la pena di familiarizzarsi con le due strategie "pure".

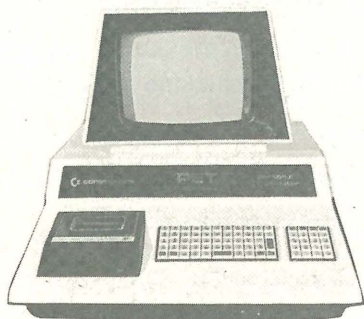
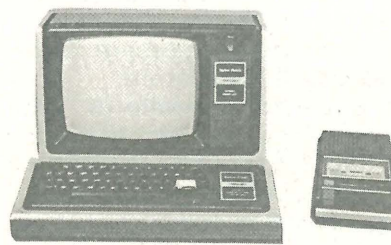
Il programma è eseguibile usando il MITS Altair 8K BASIC, revisione 4.0.

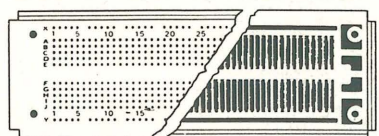
Esecuzione del programma

```
INVERSIONE

VUOI LE REGOLE? SI

QUESTO GIOCO SI CHIAMA INVERSIONE.PER GIOCARE, TU
DEVI CREARE UNA LISTA DI NUMERI (DA UNO FINO A 9)
E POI DIRMI QUANTI NUMERI(PARTENDO DA SINISTRA)VUOI
INVERTIRE. PER ESEMPIO SE LA LISTA INIZIALE E':

2 3 4 5 1 6 7 8 9

E NE VUOI INVERTIRE QUATTRO, IL RISULTATO SARA':

5 4 3 2 1 6 7 8 9

ORA SE TU NE INVERTI 5,VINCI!

1 2 3 4 5 6 7 8 9

PER ABBANDONARE, SCRIVI 0 (ZERO)


INIZIO GIOCO ... LA LISTA E':

2 4 5 1 9 6 3 7 8

QUANTE INVERSIONI? 9

8 7 3 6 9 1 5 4 2
```

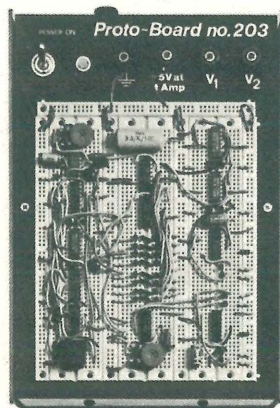
```
QUANTE INVERSIONI? 4

6 3 7 8 9 1 5 4 2

QUANTE INVERSIONI? 5

9 8 7 3 6 1 5 4 2

QUANTE INVERSIONI? 9

2 4 5 1 6 3 7 8 9

QUANTE INVERSIONI? 3

5 4 2 1 6 3 7 8 9

QUANTE INVERSIONI? 4

1 2 4 5 6 3 7 8 9

QUANTE INVERSIONI? 6

3 6 5 4 2 1 7 8 9

QUANTE INVERSIONI? 4

4 5 6 3 2 1 7 8 9

QUANTE INVERSIONI? 3

6 5 4 3 2 1 7 8 9

QUANTE INVERSIONI? 6

1 2 3 4 5 6 7 8 9

HAI VINTO IN 10 MOSSE

ANCORA (SI O NO)? NO

O.K. SPERO CHE TI SIA DIVERTITO!
```

Listing del programma

```
10 PRINT :PRINT :PRINT
100 PRINT "INVERSIONE"
110 PRINT
130 DIM A(20)
140 REM N=NUMERO DEI NUMERI
150 N=9
160 PRINT "VUOI LE REGOLE";
170 INPUT A$
180 IF A$="NO" THEN 210
190 GOSUB 710
200 REM LISTA CASUALE DA A(1) A A(N)
210 A(1)=INT((N-1)*RND(1)+2)
220 FOR K=2 TO N
230 A(K)=INT(N*RND(1)+1)
240 FOR J=1 TO K-1
250 IF A(K)=A(J) THEN 230
260 NEXT J: NEXT K
280 REM VISUALIZZAZIONE LISTA ORIGINARIA E INIZIO GIOCO
290 PRINT: PRINT "INIZIO GIOCO ... LA LISTA E':"
310 T=0
320 GOSUB 610
330 PRINT "QUANTE INVERSIONI";
340 INPUT R
350 IF R=0 THEN 520
360 IF R<=N THEN 390
370 PRINT "OOPS! TROPPO! IO POSSO INVERTIRE MASSIMO";N:GOTO 330
390 T=T+1
400 REM INVERSIONE DI R NUMERI E VISUALIZZAZIONE NUOVA LISTA
410 FOR K=1 TO INT(R/2)
420 Z=A(K)
430 A(K)=A(R-K+1)
440 A(R-K+1)=Z
450 NEXT K
460 GOSUB 610
470 REM CONTROLLO VITTORIA
480 FOR K=1 TO N
490 IF A(K)<>K THEN 330
500 NEXT K
510 PRINT "HAI VINTO IN";T;"MOSSE!":PRINT
520 PRINT
530 PRINT "ANCORA (SI O NO)";
540 INPUT A$
550 IF A$="SI" THEN 210
560 PRINT: PRINT "O.K. SPERO CHE TI SIA DIVERTITO!":GOTO 999
600 REM SUBROUTINE DI STAMPA DELLA LISTA
610 PRINT: FOR K=1 TO N: PRINT A(K);:NEXT K
650 PRINT:PRINT:RETURN
700 REM SUBROUTINE DI STAMPA DELLE REGOLE
710 PRINT:PRINT "QUESTO GIOCO SI CHIAMA INVERSIONE.PER GIOCARE, TU"
720 PRINT " DEVI CREARE UNA LISTA DI NUMERI (DA 1 FINO A";N;")"
730 PRINT "E POI DIRMI QUANTI NUMERI(PARTENDO DA SINISTRA)VUOI"
740 PRINT "INVERTIRE. PER ESEMPIO SE LA LISTA INIZIALE E':"
750 PRINT:PRINT "2 3 4 5 1 6 7 8 9"
760 PRINT: PRINT "E NE VUOI INVERTIRE QUATTRO, IL RISULTATO SARA':"
770 PRINT: PRINT "5 4 3 2 1 6 7 8 9"
780 PRINT
790 PRINT "ORA SE TU NE INVERTI 5, VINCI!"
800 PRINT: PRINT "1 2 3 4 5 6 7 8 9":PRINT
810 PRINT "PER ABBANDONARE, SCRIVI 0 (ZERO)"
820 PRINT: RETURN
999 END
```

Costruite il vostro prototipo sulle basette sperimentali CSC

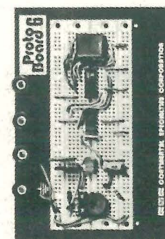
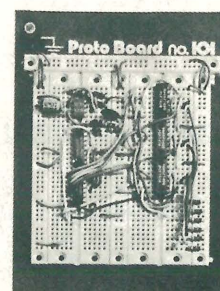
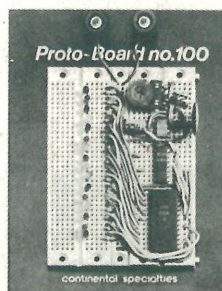
Serie EXP • Basette per esperimenti

Modello	Codice GBC	Lunghezza mm	Larghezza mm	Prezzo
EXP300	SM/4350-00	152	53	L. 14.500
EXP600	SM/4375-00	152	61	L. 15.500
EXP350	SM/4400-00	91	53	L. 7.800
EXP650	SM/4425-00	91	61	L. 8.800
EXP325	SM/4450-00	48	53	L. 3.900
EXP4B	SM/4475-00	152	25	L. 5.900

Serie QT • Basette sperimentali rapide passo 2.54 mm

Modello	Codice GBC	Lunghezza mm	Terminali	Prezzo
QT-59S	SM/4150-00	165	118	L. 17.800
QT-47S	SM/4170-00	135	94	L. 14.000
QT-35S	SM/4190-00	104	70	L. 12.000
QT-18S	SM/4210-00	61	36	L. 6.700
QT-12S	SM/4230-00	46	24	L. 5.200
QT-8S	SM/4250-00	36	16	L. 4.600
QT-7S	SM/4270-00	36	14	L. 4.200
QT-59B	SM/4290-00	165	20	L. 3.500
QT-47B	SM/4310-00	135	16	L. 3.100
QT-35B	SM/4330-00	104	12	L. 2.800

Serie PB Proto Board

Basette sperimentali con base, supporto e alimentatore

Modello	Codice GBC	Dimensioni mm	Punti di conness.	N° di IC inseribili (14 pin)	N° bas.	Tipo	Prezzo
PB-203	SM/4650-00	248x168x83	2250	24	3 4 1	QT-59S QT-59B QT-47B	L. 143.000
PB-203A	SM/4675-00	248x168x83	2250	24	3 4 1	QT-59S QT-59B QT-47B	L. 190.000

Serie PB Proto Board

Basette sperimentali con base e supporto

Modello	Codice GBC	Dimensioni mm	Punti di conness.	N° di IC inseribili (14 pin)	N° bas.	Tipo	Prezzo
PB-6	SM/4500-00	152x102x36	630	6	2 1	QT-47B QT-47S	L. 22.500
PB-100	SM/4525-00	152x114x36	760	10	2 1	QT-35S QT-35B	L. 29.000
PB-101	SM/4550-00	152x114x36	940	10	2 4	QT-35S QT-35B	L. 42.000
PB-102	SM/4575-00	187x114x36	1240	12	2 3 1	QT-47S QT-47B QT-35B	L. 56.000
PB-103	SM/4600-00	229x152x36	2250	24	3 4 1	QT-59S QT-59B QT-47B	L. 84.500
PB-104	SM/4625-00	249x203x36	3060	32	4 7	QT-59S QT-59B	L. 112.000

DISTRIBUITI IN ITALIA DALLA GBC ITALIANA - VIALE MATTEOTTI, 66 - CINISELLO BALSAMO

PREZZI IVATI

LISTINO PREZZI

PINZA LOGICA

LM1	SM/4001-00	L. **75.000**

SONDE LOGICHE

LP1	SM/4005-00	L. **80.000**
LP2	SM/4006-00	L. **47.000**
LPK1	SM/4010-00	L. **31.000**

FREQUENZIMETRI

MAX 100	SM/4025-00	L. **201.000**
MAX 50	SM/4030-00	L. **140.000**

PRESCALER

PS 500	SM/4035-00	L. **91.000**

ACCESSORI PER FREQUENZIMETRI

100 MWA	SM/4040-00	L. **6.500**
100 CA2	SM/4045-00	L. **16.000**
MMC 5	SM/4049-00	L. **9.000**

PINZE PER CIRCUITI INTEGRATI

PC 14	SM/4085-00	L. **6.600**
PC 16	SM/4090-00	L. **6.700**
PC 24	SM/4095-00	L. **12.000**
PC 40	SM/4100-00	L. **19.500**

PINZE PER CIRCUITI INTEGRATI CON CAVO

PC 14-18S	SM/4115-00	L. **13.500**
PC 14-18D	SM/4120-00	L. **23.000**
PC 16-18S	SM/4125-00	L. **15.000**
PC 16-18D	SM/4130-00	L. **25.000**

BASETTE SPERIMENTALI RAPIDE PASSO 2,54 mm.

QT 59S	SM/4150-00	L. **17.800**
QT 47S	SM/4170-00	L. **14.000**
QT 35S	SM/4190-00	L. **12.000**
QT 18S	SM/4210-00	L. **6.700**
QT 12S	SM/4230-00	L. **5.200**
QT 8S	SM/4250-00	L. **4.600**
QT 7S	SM/4270-00	L. **4.200**
QT 59B	SM/4290-00	L. **3.500**
QT 47B	SM/4310-00	L. **3.100**
QT 35B	SM/4330-00	L. **2.800**

BASETTE PER ESPERIMENTI

EXP 300	SM/4350-00	L. **14.500**
EXP 600	SM/4375-00	L. **15.500**
EXP 350	SM/4400-00	L. **7.800**
EXP 650	SM/4425-00	L. **8.800**
EXP 325	SM/4450-00	L. **3.900**
EXP 4B	SM/4475-00	L. **5.900**

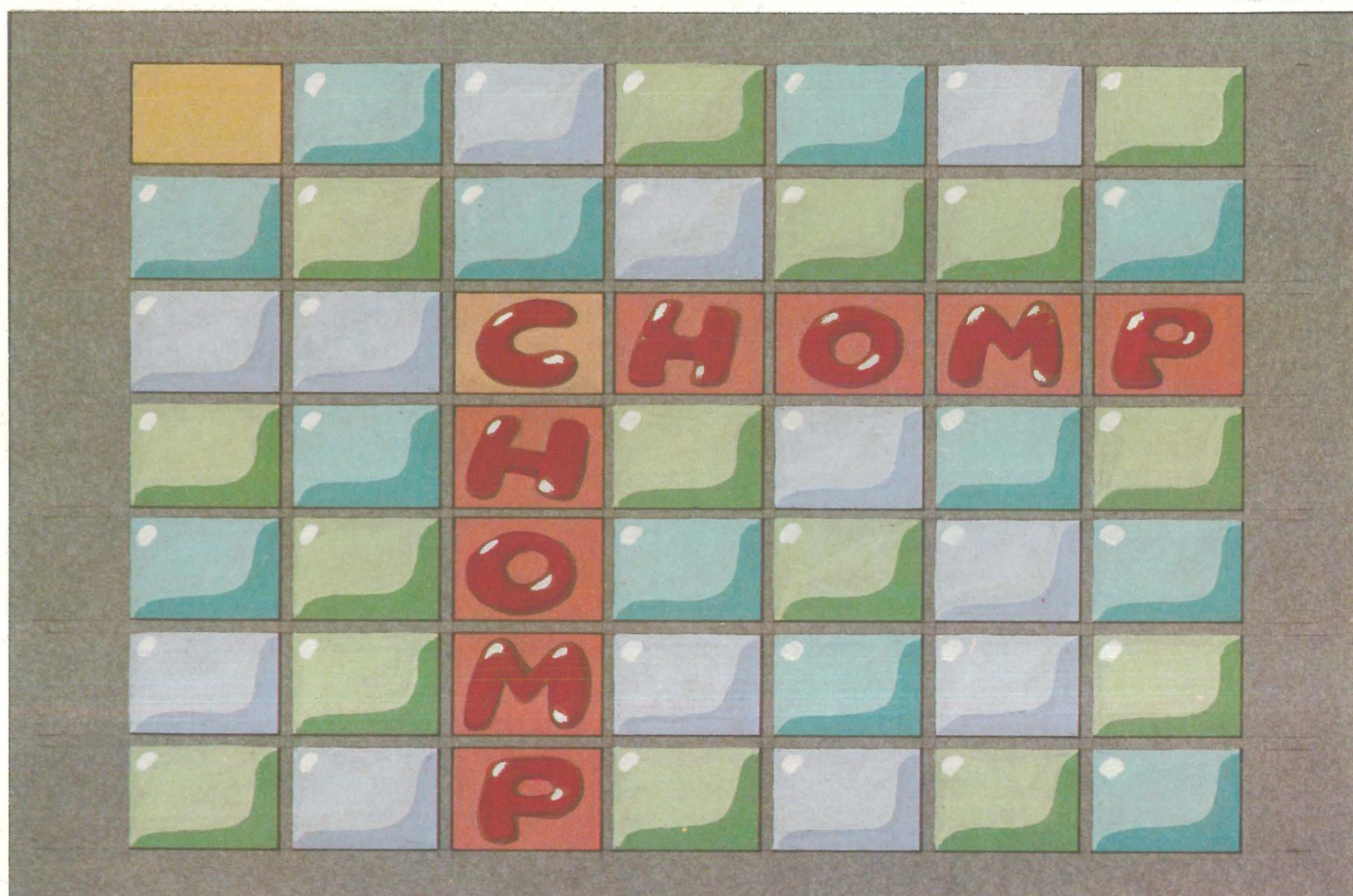
BASETTE SPERIMENTALI CON BASE E SUPPORTO

PB 6	SM/4500-00	L. **22.500**
PB 100	SM/4525-00	L. **29.000**
PB 101	SM/4550-00	L. **42.000**
PB 102	SM/4575-00	L. **56.000**
PB 103	SM/4600-00	L. **84.500**
PB 104	SM/4625-00	L. **112.000**

BASETTE SPERIMENTALI CON BASE SUPPORTO E ALIMENTATORE

PB 203	SM/4650-00	L. **143.000**
PB 203 A	SM/4675-00	L. **190.000**

PREZZI IVATI

DISTRIBUITI IN ITALIA DALLA GBC ITALIANA - VIALE MATTEOTTI, 66 - CINISELLO BALSAMO

Chomp

Tratto da "Basic Computer Games". Creative Computing Press. *

Questo programma è un adattamento di un gioco matematico originariamente illustrato da Martin Gardner nel numero di gennaio 1973 di "Scientific American".
Si stabilisce un campo di 9 caselle per 9 massimo; la prima casella in alto a sinistra è congelata; il toccarla implica la perdita della partita. I giocatori, a turno, devono cancellare un settore del campo; per farlo, digitano il numero di riga e il numero di colonna della casella posta al vertice in alto a sinistra del settore scelto. La casella così individuata, e tutte quelle che si trovano al di sotto e alla destra di essa, scompaiono.
Può partecipare al gioco un numero qualsiasi di giocatori; il calcolatore non partecipa al gioco, ma ha soltanto il ruolo di arbitro. Il gioco permette di attuare delle strategie interessanti comunque; ma se i giocatori sono più di due, diventa un'autentica sfida.
Il programma è eseguibile usando il MITS Altair 8K BASIC, revisione 4.0.

Esecuzione del programma

```
QUESTO GIOCO SI CHIAMA CHOMP
VUOI LE REGOLE (1=SI, 0=NO)? 1
CHOMP E' PER UNO O PIU' GIOCATORI

ECCO COME APPARE IL CAMPO (QUESTO E' 5*7):

         1 2 3 4 5 6 7 8 9
  1      P * * * * * *
  2      * * * * * * *
  3      * * * * * * *
  4      * * * * * * *
  5      * * * * * * *


IL CAMPO E' FORMATO DA R RIGHE E C COLONNE.
ALL'INIZIO ENTRA R E C. IN ALTO A SINISTRA
C'E' UNA CASELLA VIETATA (P).
CHI ACCEDE ALLA CASELLA VIETATA PERDE.
PER GIOCARE DIGITA IL NUMERO DI RIGA E DI COLONNA
DI UNA CASELLA DEL CAMPO. TUTTE LE CASELLE POSTE
SOTTO E A DESTRA DI QUESTA SCOMPAIONO (OVVERO CHOMP!).
E' ERRATO CHIAMARE CASELLE ELIMINATE O FUORI DELLA
ORIGINALE DIMENSIONE DEL CAMPO.
```

* Distribuito in Italia dalla Jackson Editrice s.r.l.

```
INIZIO GIOCO

QUANTI GIOCATORI? 2
QUANTE RIGHE? 8
QUANTE COLONNE? 7

       1 2 3 4 5 6 7 8 9
 1     P * * * * * *
 2     * * * * * * *
 3     * * * * * * *
 4     * * * * * * *
 5     * * * * * * *
 6     * * * * * * *
 7     * * * * * * *
 8     * * * * * * *

GIOCATORE   1
COORDINATE DELLA CASELLA (RIGA,COLONNA)? 5,6

       1 2 3 4 5 6 7 8 9
 1     P * * * * * *
 2     * * * * * * *
 3     * * * * * * *
 4     * * * * * * *
 5     * * * * *
 6     * * * * *
 7     * * * * *
 8     * * * * *

GIOCATORE   2
COORDINATE DELLA CASELLA (RIGA,COLONNA)? 3,2

       1 2 3 4 5 6 7 8 9
 1     P * * * * * *
 2     * * * * * * *
 3     *
 4     *
 5     *
 6     *
 7     *
 8     *

GIOCATORE   1
COORDINATE DELLA CASELLA (RIGA,COLONNA)? 4,4
ERRORE. HAI CHIAMATO UNA CASELLA INESISTENTE!
GIOCATORE   1
COORDINATE DELLA CASELLA (RIGA,COLONNA)? 2,2

       1 2 3 4 5 6 7 8 9
 1     P * * * * * *
 2     *
 3     *
 4     *
 5     *
 6     *
 7     *
 8     *

GIOCATORE   2
COORDINATE DELLA CASELLA (RIGA,COLONNA)? 1,2

       1 2 3 4 5 6 7 8 9
 1     P
 2     *
 3     *
 4     *
 5     *
 6     *
 7     *
 8     *

GIOCATORE   1
COORDINATE DELLA CASELLA (RIGA,COLONNA)? 2,1

       1 2 3 4 5 6 7 8 9
 1     P
 2
 3
 4
 5
 6
 7
 8

GIOCATORE   2
COORDINATE DELLA CASELLA (RIGA,COLONNA)? 1,1
HA PERSO IL GIOCATORE  2

ANCORA (1=SI; 0=NO)? 0
```

```
10 PRINT: PRINT : PRINT
50 DIM A(10,10)
120 PRINT " QUESTO GIOCO SI CHIAMA CHOMP"
130 PRINT "VUOI LE REGOLE (1=SI, 0=NO)";
140 INPUT R
150 IF R=0 THEN 340
160 F=1
170 R=5
180 C=7
190 PRINT "CHOMP E' PER UNO O PIU' GIOCATORI."
200 PRINT
210 PRINT "ECCO COME APPARE IL CAMPO (QUESTO E' 5*7):"
220 GOSUB 540
230 PRINT
240 PRINT "IL CAMPO E' FORMATO DA R RIGHE E C COLONNE."
250 PRINT "ALL'INIZIO ENTRA R E C.IN ALTO A SINISTRA"
260 PRINT "C'E' UNA CASELLA VIETATA (P)."
270 PRINT "CHI ACCEDE ALLA CASELLA VIETATA PERDE."
280 PRINT "PER GIOCARE DIGITA IL NUMERO DI RIGA E DI COLONNA"
290 PRINT "DI UNA CASELLA DEL CAMPO. TUTTE LE CASELLE POSTE"
300 PRINT "SOTTO E A DESTRA DI QUESTA SCOMPAIONO (OVVERO CHOMP!)."
310 PRINT "E' ERRATO CHIAMARE CASELLE ELIMINATE O FUORI DELLA"
320 PRINT "ORIGINALE DIMENSIONE DEL CAMPO."
330 PRINT
340 PRINT "INIZIO GIOCO"
350 REM
360 F=0
370 FOR I=1 TO 10
372 FOR J=1 TO 10
375 A(I,J)=0
377 NEXT J
379 NEXT I
380 PRINT
390 PRINT "QUANTI GIOCATORI";
400 INPUT P
410 I1=0
420 PRINT "QUANTE RIGHE";
430 INPUT R
440 IF R <= 9 THEN 470
450 PRINT "TROPPE RIGHE(9 E' IL MASSIMO).ORA, ";
460 GOTO 420
470 PRINT "QUANTE COLONNE";
480 INPUT C
490 IF C <= 9 THEN 530
500 PRINT "TROPPE COLONNE(9 E' IL MASSIMO).ORA, ";
510 GOTO 470
530 PRINT
540 FOR I=1 TO R
550 FOR J=1 TO C
560 A(I,J)=1
570 NEXT J
580 NEXT I
590 A(1,1)=-1
600 REM VISUALIZZA IL CAMPO
610 PRINT
620 PRINT TAB(7);"1 2 3 4 5 6 7 8 9"
630 FOR I=1 TO R
640 PRINT I;TAB(7);
650 FOR J=1 TO C
660 IF A(I,J)=-1 THEN 700
670 IF A(I,J)=0 THEN 720
680 PRINT "* ";
690 GOTO 710
700 PRINT "P ";
710 NEXT J
720 PRINT
730 NEXT I
740 PRINT
750 IF F=0 THEN 770
760 RETURN
770 REM CHIAMA A TURNO I GIOCATORI
780 LET I1=I1+1
790 LET P1=I1-INT(I1/P)*P
800 IF P1 <> 0 THEN 820
810 P1=P
820 PRINT "GIOCATORE ";P1
830 PRINT "COORDINATE DELLA CASELLA (RIGA,COLONNA)";
840 INPUT R1,C1
850 IF R1<1 THEN 920
860 IF R1>R THEN 920
870 IF C1<1 THEN 920
880 IF C1>C THEN 920
890 IF A(R1,C1)=0 THEN 920
900 IF A(R1,C1)=-1 THEN 1010
910 GOTO 940
920 PRINT "ERRORE.HAI CHIAMATO UNA CASELLA INESISTENTE!"
930 GOTO 820
940 FOR I=R1 TO R
950 FOR J=C1 TO C
960 A(I,J)=0
970 NEXT J
980 NEXT I
990 GOTO 610
1000 REM FINE DEL GIOCO RILEVATA ALLA LINEA 900
1010 PRINT "HA PERSO IL GIOCATORE ";P1
1020 PRINT
1030 PRINT "ANCORA (1=SI; 0=NO)";
1040 INPUT R
1050 IF R=1 THEN 340
1060 END
```

Il Nanocomputer NBZ80:
Prima applicazione
Utilizzo del display della tastiera come periferica di uscita

di **Aldo Cavalcoli** - *MIPRO s.r.l.*

Premessa

Questo secondo articolo della serie NBZ80 presenta una prima semplice applicazione del Nanocomputer, in particolare si vedrà come utilizzare il display del miniterminale come periferica di uscita per la visualizzazione di un messaggio scelto dall'utente. Data la semplicità del software necessario, verrà effettuata un'accurata indagine delle modalità operative dello NBZ80 e dell'organizzazione software del monitor residente NC-Z.

Il Monitor NC-Z

Il Monitor NC-Z, presente su EPROM sulla scheda NBZ80, contiene un certo numero di subroutine utilizzabili direttamente dall'utente, nel modo seguente:
CALL XXXX
dove XXXX è l'indirizzo assoluto della subroutine.
Le subroutine utilizzabili, con l'indirizzo di chiamata entro parentesi, sono:

KBSCAN	(F8DB)	= scansione, tastiera
DISPL	(F909)	= gestione dei display a 7 segmenti del miniterminale
NULL	(F96E)	= invio di un carattere NULL verso la TTY/CASSETTA
TTYO	(F970)	= invio di un carattere verso la TTY/CASS
TTYI	(F9AA)	= ingresso di un carattere da TTY/CASS
TTYI1	(F9C6)	= ingresso di un carattere da TTY/CASS per nastri formattati in record
BAUD	(F9F2)	= ritardo per generare il baudrate
BAUDHF	(F9FE)	= come la precedente, ma con ritardo 1/2
ASCII	(FA10)	= invio dell'equivalente ASCII di un byte binario (HL), visto come due caratteri esadecimali, verso la TTY
ASCIB	(FA12)	= come la precedente, ma con il byte in B
BYTE	(FA2D)	= lettura di due caratteri ASCII e loro conversione in un byte in E
CONVDI	(FA7C)	= conversione di due parole (di 16 bit ciascuna), dato ed indirizzo per il display del miniterminale, in codice 7-segmenti per la visualizzazione con DISPL

Nella presente applicazione verrà utilizzata solo la subroutine DISPL.

Il display del miniterminale del Nanocomputer

Il miniterminale del Nanocomputer permette la visualizzazione di dati ed indirizzi su un display formato da otto digit a 7 segmenti. Ricordiamo di seguito le notazioni relative ai singoli segmenti di un display a sette segmenti:

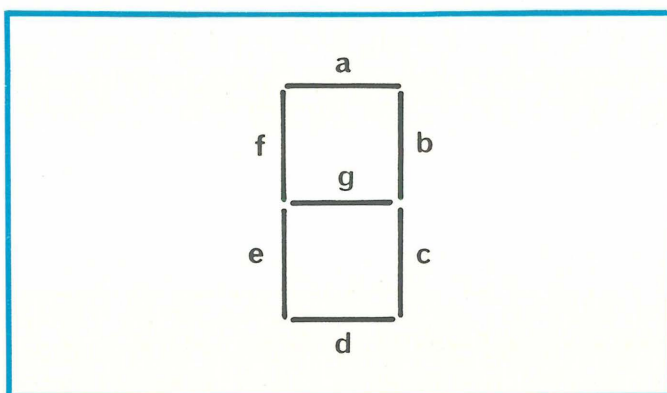



Fig. 1 - Un display a 7 segmenti

con un display a 7 segmenti, si possono allora rappresentare tutti i numeri esadecimali, accendendo i segmenti necessari, e cioé:

Digit esadecimali	Combinazione dei segmenti
0 →	a, b, c, d, e, f
1 →	b, c
2 →	a, b, d, e, g
3 →	a, b, c, d, g
4 →	b, c, f, g
5 →	a, c, d, f, g
6 →	a, c, d, e, f, g
7 →	a, b, c
8 →	a, b, c, d, e, f, g
9 →	a, b, c, f, g
A →	a, b, c, e, f, g
b →	c, d, e, f, g
C →	a, d, e, f
d →	b, c, d, e, g
E →	a, d, e, f, g
F →	a, e, f, g

Con riferimento a quanto detto, volendo usare il display come periferica di uscita, è indispensabile poter accendere a proprio piacere i singoli segmenti di ciascun digit, al fine di ottenere la visualizzazione di quanto si desidera.
Questo può avvenire utilizzando la DISPL.
Questa subroutine, che descriveremo tra poco, inviando verso il display determinati byte, comanda l'accensione di questo o quel segmento.

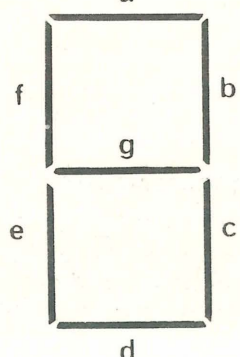
Sarà compito del programmatore scrivere un programma che, utilizzando le proprietà della DISPL e le convenzioni di pilotaggio del display, mette in uscita il messaggio voluto.
Le convenzioni che servono con la DISPL sono date nella tabella seguente.

BIT	SEGMENTO ACCESO
0	Nessuno (N)
1	g
2	f
3	e
4	d
5	c
6	b
7	a

Tabella 1 - Codifica bit/segmento per la subroutine DISPL

Dalla tabella precedente, se si vuole rappresentare la lettera L, cioè:

si dovranno accendere i segmenti: d, e, f
Dalla corrispondenza BIT/SEGMENTO, si ricava:

cioé:
00011100 = 1C in esadecimale.
La DISPL deve porre perciò in uscita verso il display il byte 1C per avere accesi d, e, f e quindi per avere rappresentata la lettera L.

La subroutine DISPL

La subroutine DISPL viene qui presentata per completezza di esposizione e come riferimento per il lettore. Non si entra quindi in dettaglio sull'hardware della scheda NBZ80.
Per quanti fossero interessati ad approfondire l'argomento, si rimanda al Nanobook Vol.III, di prossima pubblicazione da parte della Jackson Italiana Editrice. La DISPL utilizza, per pilotare il display, il contenuto di 10 locazioni RAM, contenenti le informazioni relative ai digit di indirizzo e dati del display e relative agli indicatori luminosi del miniterminale.
Queste locazioni sono così suddivise:

ADD7 : i 4 codici ad un byte per il pilotaggio dei 4 digit della zona "indirizzo" del display
DATA7 : i 4 codici ad un byte per il pilotaggio della zona "dati" del display
LEDH : i 2 byte i cui bit indicano quale indicatore luminoso (led) deve essere acceso (14 Led in tutto)

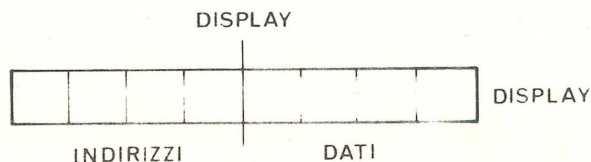
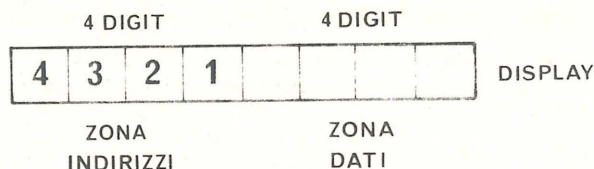



Fig. 2 - Le due zone del display

Come già visto nel precedente articolo, gli indicatori luminosi servono per selezionare "l'entità" alla quale si vuole accedere (memoria, registro A, B, C, D, ... ecc.).
Per quanto concerne il gruppo di locazioni ADD7, che inizia all'indirizzo OFBA in RAM, vale quanto segue:

ADD7 : codice esadecimale dei 4 bit più significativi del byte di indirizzo HI (high, alto), digit 4 del display
ADD7 + 1: codice esadecimale dei 4 bit meno significativi del byte di indirizzo HI, digit 3
ADD7 + 2: codice esadecimale dei 4 bit più significativi del byte di indirizzo LO (low, basso) digit 2
ADD7 + 3: codice esadecimale dei 4 bit meno significativi del byte di indirizzo LO, digit 1

Stessa situazione vale per le locazioni DATA7 (DATA7, DATA7 + 1, DATA7 + 2, DATA7 + 3).
A questo punto dovreste aver capito che per visualizzare nel digit più a sinistra del display una lettera (diciamo la L), occorre introdurre in OFBA il byte 1C (si veda la tabella di codifica bit/segmento).
Si tenga presente che nel Nanocomputer la conversione in codice-7 segmenti è eseguita automaticamente dalla subroutine CONVDI.

Spiegazione dell'applicazione

1) Programma
Scopo dell'applicazione è quello di visualizzare sul display un messaggio di 20 caratteri a piacere, purché compatibili con la possibilità rappresentativa dei display a 7 segmenti (non sarà certo possibile rappresentare una R, ma si potrà rappresentare una F).
Proponiamo un programma di visualizzazione di un messaggio di 20 caratteri e la sua successiva spiegazione e debug.

```
0200   210301   DUPL:  LD   HL, MES
0203   111701          LD   DE, MESDU
0206   010700          LD   BC, 07H
0209   EDBO            LDIR
020B   3E80     POINT: LD   A, 80H
```

```
020D  320001           LD   (COUNT), A
0210  2A0101           LD   HL, (PMESCO)
0213  11BAOF           LD   DE, ADD7
0216  010800           BC,  08H
0219  EDBO             LDIR
021B  CD09F9   DYOUT:  CALL DISPL
021E  3A0001           LD   A, COUNT
0221  3D               DEC  A
0222  FE00             CP   0
0224  320001           LD   (COUNT), A
0227  C21B02           JP   NZ, DYOUT
022A  3A0101   SHIFT:  LD   A, (PMESCO)
022D  3C               INC  A
022E  FE17     TSHIFT: CP   MESDU
0230  C23502           JP   NZ, AVANTI
0233  3E03             LD   A, MES
0235  320101   AVANTI: LD   (PMESCO), A
0238  C30B02           JP   POINT
```

2) Mappa di memoria ed inizializzazione

Come si può vedere, il programma parte dalla locazione di memoria 0200.
Vi sono delle inizializzazioni preliminari da eseguire; vale la seguente mappa di memoria:

Zona	Indirizzo
NON USATO	0000
COUNT	0100
PMESCO LOW	0101
PMESCO HIGH	0102
MES Zona di memorizz. del messaggio	0103
	0116
MESDU 7 LOCAZIONI	0117
: NON USATO	011E
PROGRAMMA	0200
	0238
: NON USATO	
ADD 7	0FBA

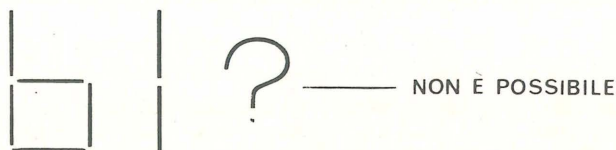
La locazione di memoria 0100, indicata con COUNT, conterrà un valore esadecimale che determina di fatto la velocità di scorrimento del messaggio sul display.
Nel nostro caso, in COUNT vi sarà 80, a seguito della sequenza di istruzioni:

```
LD 1,80H
LD (COUNT), A
```

In PMESCO (HIGH e LOW) occorre mettere l'indirizzo di partenza della tabella in cui è memorizzato il messaggio. Facendo partire la tabella da 0103, sarà quindi:

03 → PMESCO LOW, cioè in 0101
01 → PMESCO HIGH, cioè in 0102

Si tenga presente che questa allocazione è del tutto arbitraria. Successivamente, scelto il messaggio, si deve caricare la sequenza di byte corrispondente nella zona da 0103 a 0116.

3) Messaggio
Si scelga ad esempio il seguente messaggio: BIT.
Questo messaggio di 3 lettere occuperà solo 3 byte.
La codifica per l'opportuna accensione dei segmenti si ottiene dalla tabella 1 e dalla seguente situazione reale di rappresentazione:

Il punto di domanda sta ad indicare che con i 7 segmenti non si può rappresentare una T.
Potremmo in effetti rappresentare una "t" minuscola. Come? A voi la risposta.
Questo, solo per ricordare che si è di fatto limitati in quello che si vuole rappresentare.
Proviamo con il messaggio SGS.
La rappresentazione è

quindi possibile.
La codifica è

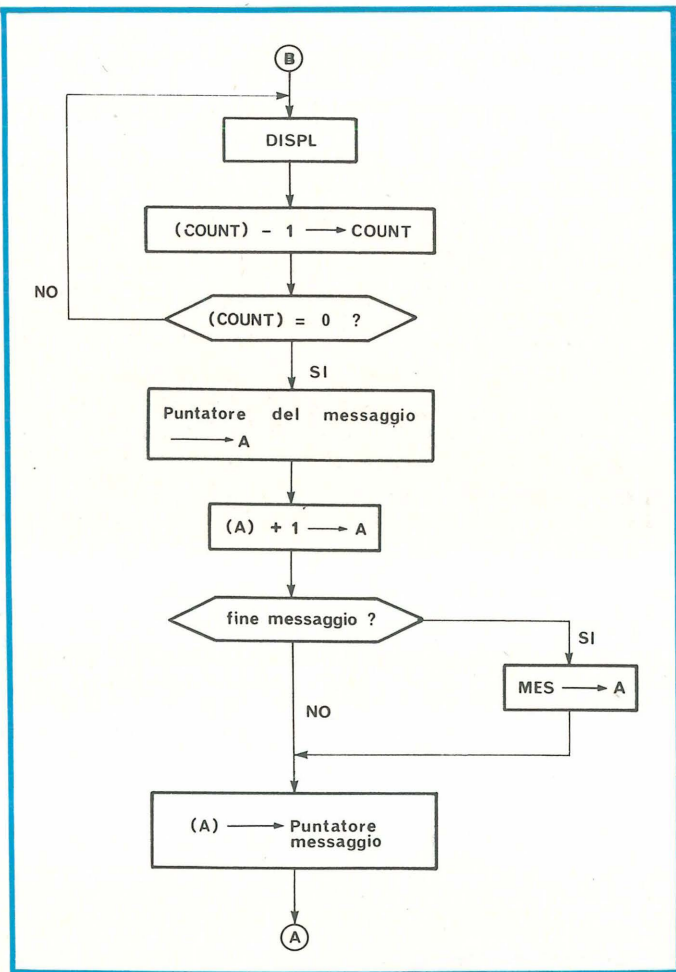
S → a, c, d, f, g = 10110110 = B6
G → a, c, d, e, f, = 10111100 = BC
S → a, c, d, f, g = 10110110 = B6

Quindi sarà:

B6 → 0103
BC → 0104
B6 → 0105

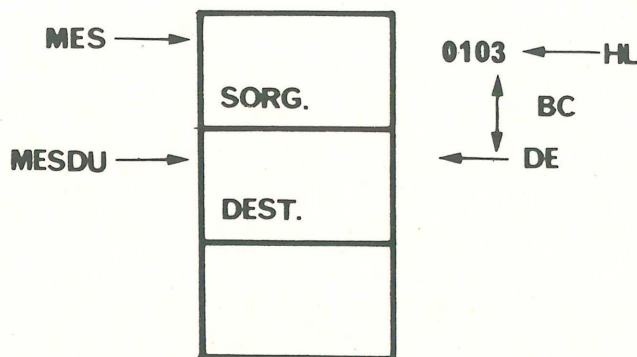
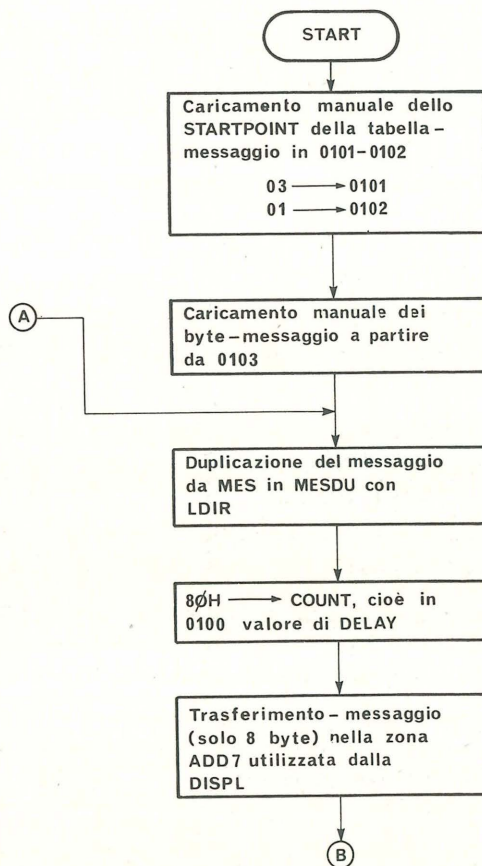
Da 0106 a 0116, le altre locazioni di memorizzazione del messaggio, si carica 00, in modo da spegnere il display in corrispondenza, per avere solo "SGS", che scorre sul display stesso.

4) Caricamento del programma
Utilizzando il tasto "freccia a destra" o "freccia a sinistra" si posiziona il selettore luminoso su MEM.
Si visualizza la prima locazione di memoria del programma, cioè 0200, e si inizia a caricare il programma. La sequenza è:

1) Led su MEM
2) 0200/LA (si digita 0200 e poi si preme il tasto LA, cioé Load Address)
3) 21/ST (si inizia a caricare il programma, con il 1° byte 21, seguito dal tasto ST, cioé STORE)
4) 03/ST
ecc.
ecc.

n) 02/ST (ultimo byte del programma)
Si introducano ora i dati di inizializzazione:
a) PMESCO
1) Led su MEM
2) 0101/LA
3) 03/ST
4) 01/ST
b) MES
1) Led su MEM (probabilmente c'è già)
2) B6/ST
3) BC/ST
4) B6/ST

5) Debug del programma
Nella messa a punto (debug) del programma, verranno usate le possibilità del Monitor NC-Z.
Prima di tutto la funzione SS (Single-Step).

La sequenza è:

1) Led su PC
2) 0200/ST
3) SS; viene eseguita la LD HL, MES
Ci aspettiamo quindi che MES, cioè 0103 vada in HL.
4) Con il tasto "freccia..." si metta il selettore su HL: appare 0103
5) SS; viene eseguita la LD DE,MESDU.
Ci aspettiamo che MESDU, cioè 0117 vada in DE
6) Led su DE: appare 0117
7) SS; viene eseguita la LD BC,07H
8) Led su BC: appare 0007
9) Restando con il Led su BC, si esegua la LDIR; ogni volta che si preme SS, si esegue un ciclo della LDIR, cioè viene effettuato un trasferimento come da figura, e BC si decrementa di 1, come appunto vuole la modalità di esecuzione della LDIR.

In questo modo, passo-passo, si può vedere l'effetto di ogni singola istruzione.
Un'altra possibilità è quella di eseguire una parte del programma e poi arrestare il microcomputer.
Questo avviene usando il breakpoint.

Si voglia eseguire il programma da DUPL fino a SHIFT, cioé fino alla locazione 022A.
La sequenza è:
1) RESET
2) Led su PC, 0200/ST
3) Si preme BRK per entrare in Breakpoint mode
4) 022A/LA, cioé si inizializza il Breakpoint a 022A
5) BRK, per uscire dal Breakpoint mode
6) GO
Dopo aver premuto il tasto di GO, vedete apparire per breve tempo la scritta SGS, poi sul display appare: 022A 3A.
Cioé, eseguita una visualizzazione, ci si arresta in 022A come desiderato, prima che la scritta venga fatta scorrere.
È chiaro che a questo punto, sia con SS che con BRK potete realizzare il debug più vario da voi scelto.
Per togliere il breakpoint, basta premere RESET, per cui per vedere la circolazione del messaggio basta fare:
1) RESET
2) Led su PC, 0200/ST
3) GO
Sul display appare SGS come messaggio che scorre sul display stesso.

PERIFERICHE

Terminale video Hazeltine mod. 1552

L'Hazeltine, la ben nota costruttrice delle più famose serie 1500 e 1400, ha annunciato nel corso della National Computer Conference di New York un altro terminale video: il modello 1552.
Il modello 1552 rappresenta qualcosa di più di un semplice terminale video poiché per la prima volta la Hazeltine, e può voler significare una precisa tendenza, ha prodotto un terminale perfettamente compatibile con un prodotto già esistente e affermato quale il modello VT-52 della Digital Equipment Corporation.
Oltre ad avere tutte le caratteristiche del modello della DEC, il mod. 1552 è dotato di alcuni dispositivi standard che lo rendono addirittura superiore al VT-52.
Il mod. 1552 sarà disponibile per consegne in Italia a partire da Settembre 1979.
Tutta la linea HAZELTINE è rappresentata in esclusiva per l'Italia dalla Segi Servizi Generali per l'Informatica.

SEGI-MILANO

BASF 6106: il più mini dei minifloppy

Alto 53,5 mm. invece dei tradizionali 82,5 mm., il BASF 6106 minifloppy driver è sicuramente il più impaccato esistente sul mercato. Infatti tutta la logica di controllo è su una singola scheda e la meccanica adottata è delle più raffinate sia per miniaturizzazione che per prestazioni. Il risultato finale è un driver veramente MINI con 125 K bytes invece dei tradizionali 100 Kbytes (cinque tracce in più), con dei tempi di accesso traccia tre volte più veloci dei tradizionali (12 m/sec. TK-TK) che pesa soltanto 1,4 Kg. e consuma solo 10-15 watt. Il BASF 6106 è distribuito in Italia dalla MICROLEM data.

MICROLEM data - Vimercate (MI)

Completa la famiglia Printronix

La "Line Printer" Printronix P-600 è oggi finalmente disponibile anche in Italia.
Lo stesso principio tecnologico di base, una stampa cioé con martelletti a matrice 7 × 9 o 9 × 9, delle sorelle minori P-150 e P-300 assicura anche alla P-600 la stessa collaudata qualità, affidabilità e versatilità che già più di duecento utenti italiani hanno con estremo favore constatato sulle loro P-150 e P-300.
Anche la P-600 può lavorare sia in plot-mode che in block-mode ed è dotata delle interfacce parallele Centronics o Data Products ed in opzione di Controller per Data General, DEC ed altri oltre alla possibilità di collegamento in RS232 (V24).
La PRINTRONIX è rappresentata in esclusiva per l'Italia da: SEGI Servizi Generali per l'Informatica

SEGI - MILANO

MICROCOMPUTERS

L'LSI-11/23: una nuova generazione di "micro"

La Digital ha annunciato l'LSI-11/23, un nuovo potente membro della famiglia di microelaboratori a 16 bit LSI-11. Con l'introduzione di questo prodotto le possibilità offerte dai micro Digital sono notevolmente aumentate. Come i modelli LSI-11/2 e PDP-11/03, il nuovo 11/23 utilizza i moduli altamente compatti a doppi connettori da 13,2 cm. × 22,8 cm. L'unità centrale a 16 bit contiene: il processore, circuiti potenziati di interfacciamento al bus LSI-11, uno zoccolo per l'unità di "memory management" e degli zoccoli ausiliari per opzioni come l'unità per operazioni in virgola mobile. L'LSI-11/23 incorpora su una singola scheda la potenza dei minielaboratori appartenenti alla fascia media della famiglia PDP-11, pur mantenendo la caratteristica di "micro" elaboratore.
L'ampia gamma di software sviluppato dalla Digital sugli elaboratori della famiglia PDP-11 è completamente compatibile con il nuovo micro che nella famiglia LSI-11 viene ad occupare il gradino più alto, mentre il PDP-11/23 è un'estensione del PDP-11/03. Entrambi i prodotti rappresentano la quarta generazione della famiglia PDP-11 e sono compatibili con tutti gli altri modelli. Comparati ai loro rispettivi predecessori LSI-11/2 e PDP-11/03, i nuovi modelli presentano una capacità di memoria ed un numero di livelli di interrupt quadriplicato e, a seconda della configurazione, la velocità viene aumentata da due a quattro volte.
L'LSI-11/23 e il PDP-11/23 incorporano il set di istruzioni del mini PDP-11/34 e supportano i sistemi operativi multiutente RSX11-M e RSX11-S multitask, oltre all'RT11 e ai vari linguaggi di alto livello come il BASIC e il FORTRAN IV.

DIGITAL EQUIPMENT - Cinisello B. (MI)

Microcomputer su singola scheda

La Motorola Microsystems ha immesso sul mercato due moduli microcomputer a singola scheda -

l'M68MM01B e l'M68MM01B1A. Le schede, che si aggiungono alla produzione già esistente di moduli della Motorola, offrono una serie di nuove caratteristiche, tra cui un timer programmabile per controlli di processo.
L'M68MM01B è un microcomputer stand-alone che possiede tutta la potenza di elaborazione e di controllo di una MPU MC6802, con un proprio circuito di clock a 128 byte di RAM statica. Oltre a ciò, il modulo contiene socket sufficienti per 4K di EPROM o ROM, un Peripheral Interface Adapter (PIA) per il trasferimento parallelo dei dati, ed un timer programmabile (PTM).
Necessita di una sola alimentazione + 5V.
L'M68MM01B1A costituisce una versione più "densa" dell'M68MM01B, con 256 byte addizionali di RAM statica; è inoltre in grado di effettuare il refresh della memoria dinamica off-board e possiede una Asynchronous Communications Interface Adapter (ACIA) con circuiti di interfaccia RS-232C, un circuito di interfaccia per cassetta a nastro, nonché i bus driver per i segnali dei bus degli indirizzi, dei dati e di controllo. Per realizzare una più ampia capacità di memoria e di I/0 al modulo in oggetto si possono aggiungere dispositivi addizionali, esterni alla scheda M68MM01B1A, al contrario di quanto accade con l'M68MM01B, che non ha disponibili i segnali relativi agli indirizzi, ai dati ed al controllo, per l'interfacciamento con altri moduli.
Inoltre, il modulo M68MM01B1A è compatibile a livello bus con l'EXORciser M6800. Tale versatilità offre all'utente i mezzi necessari per sviluppare un proprio sistema e quindi effettuare il debug del software, e ricercare gli eventuali malfunzionamenti relativi all'hardware prodotto.
Pertanto l'utente può sfruttare a proprio vantaggio i moduli opzionali di debug, di memoria e di I/0 dell'EXORciser.

MOTOROLA - MILANO

Disponibile la piastra di sviluppo per lo Z8000 Zilog

È stata recentemente resa disponibile una piastra (chiamata Z8000 Development Module) per lo sviluppo di progetti hardware e software basati sullo Z8000.
Di seguito riportiamo brevemente le caratteristiche salienti di tale nuova piastra, disponibile presso la Zelco, rappresentante esclusiva per l'Italia della Zilog:

- CPU Z8000 (versione a 48 o a 40 pin a scelta)
- 2 Kword (4 Kbyte) di EPROM, espandibili a 8
- 16 Kword (32 Kbyte) di RAM, espandibili a 32
- doppia interfaccia seriale (110 baud - 19,2 kbaud)
- 32 linee programmabili di I/0 con capacità di handshake
- 4 contatori/temporizzatori programmabili a 8 bit
- area per "wire-wrap" (circa 115 cm^2)
- frequenza del clock selezionabile per mezzo di jumper

La piastra di sviluppo Z8000 è un microcalcolatore su scheda specificamente progettato per aiutare l'utente nella valutazione e nella messa a punto hardware e software di progetti impieganti la CPU Z8000 della Zilog.
Oltre alle abbondanti risorse sopra elencate, la piastra mette a disposizione un'area a "wire-wrap" per ospitare interfacce o circuiti particolari richiesti dall'utente.
Lo "Z8000 Development Module" è un sistema per lo sviluppo di prototipi con alte prestazioni e tutta la flessibilità necessaria per soddisfare qualsiasi esigenza degli utenti. Citiamo fra le altre le possibilità di scegliere, per mezzo di jumper e di micro-switch, le seguenti opzioni:

- frequenza di clock di 2,5 o 3,9 MHz
- impiego delle EPROM tipo 2708, 2716 o 2732
- uso di memorie RAM 4K × 1 o 16K × 1
- interfaccia seriale per modem, terminale o telescrivente
- scelta degli indirizzi di I/0
- scelta della frequenza di trasmissione (110 baud - 19,2 Kbaud)

La piastra di sviluppo per Z8000 comunica con l'esterno per mezzo di due interfacce seriali RS232.

Una di queste è sempre collegata ad un terminale che funge da "console" dell'insieme; l'altra può essere connessa ad un calcolatore o ad un sistema di sviluppo Zilog (o a qualsiasi altro tipo di calcolatore). Tale caratteristica permette il trasferimento dei programmi dal calcolatore principale (il cosiddetto "host computer") direttamente alla piastra di sviluppo.
Per questo tipo di collegamento non è necessario alcun tipo di hardware aggiuntivo: basta soltanto il cavo di connessione.
Un programma di Monitor, contenuto in 4 Kbyte di EPROM, mette a disposizione tutti i comandi necessari per il Debug, il controllo dell'I/0, l'interfacciamento e il trasferimento dei programmi con il sistema "madre".

ZELCO - MILANO

COMPUTERS

Home computer Texas Instruments

La Texas Instruments ha finalmente annunciato il suo primo home computer in occasione del Consumer Electronics Show di Chicago.
Il sistema, denominato TI 99/4, è stato sviluppato attorno al microprocessore a 16 bit 9900 ed è fornito di 16K di RAM, tastiera, monitor TV a colori di 13 pollici e di BASIC esteso per la programmazione.
Utilizzando una serie di moduli "allo stato solido" intercambiabili (Solid State Software Command Module TM), contenenti programmi preregistrati, il TI-99/4 permette una vasta gamma di applicazioni. L'operatività del sistema è semplice: per accedere ad un programma, tutto ciò che si deve fare è inserire lo specifico modulo e premere pochi tasti sulla tastiera, quindi il monitor visualizzerà le istruzioni necessarie per la continuazione.
I moduli annunciati comprendono: gestione del bilancio familiare, giochi didattici per bambini, grammatica di base, dietetica, football americano, grafici, dimostrazione e diagnostica.
I moduli preregistrati Texas Instruments per il TI-99/4 sono ben diversi dai programmi registrati su nastro comunemente in uso. I moduli Texas Instruments, ciascuno dei quali consiste di 5 memorie a sola lettura (ROM), permettono all'utilizzatore un accesso immediato ai programmi senza dover aspettare che il contenuto del nastro venga trasferito nella memoria del

computer, processo quest'ultimo che può richiedere anche parecchi minuti. Tra le periferiche disponibili in un immediato futuro, sono stati annunciati stampante, memoria a disco e un sintetizzatore vocale allo stato solido (Solid State Speech TM).
Utilizzando la stessa tecnologia d'avanguardia del gioco didattico "Speak & Spell" della Texas Instruments, il sintetizzatore contiene un vocabolario di 200 parole che può essere usato per ottenere dai programmi messaggi e risultati "parlati".

TEXAS-INSTRUMENTS - RIETI

Computer da tavolo HP

I notevoli miglioramenti apportati al Sistema 45 della Hewlett-Packard hanno fatto raggiungere a questo computer da tavolo un nuovo e più alto livello di prestazioni.

Una memoria centrale espandibile fino a 449 Kbytes interamente disponibile per i programmi utenti, una riduzione di oltre il 50% del costo dell'ampliamento della memoria, modifiche al sistema operativo che lo rendono più veloce e facile da usare, maggiore potenziale di crescita, nuova disponibilità di tastiere "nazionalizzate", set di caratteri per 5 differenti gruppi linguistici e adattabilità ad una nuova stampante e ad un nuovo disco opzionali, sono le caratteristiche principali del Sistema 45B.
L'HP ha anche introdotto 45 packages di software per il Sistema 45B. Di questi, 17 sono adattamenti da altri sistemi mentre 28 sono completamente nuovi e comprendono applicazioni nel campo dell'ingegneria civile, della gestione aziendale, della progettazione di circuiti elettrici, etc.
La facilità di funzionamento è il risultato delle modifiche apportate al sistema operativo. Molte di queste modifiche sono estremamente sofisticate, come il trasferimento dell'indice delle cartucce a nastro nella memoria principale e l'aumento del numero massimo di numeri di linea, che ora è di 32.766.
Gli alloggiamenti per le ROM opzionali sono stati portati da 8 a 16.
Il Sistema 45B dispone inoltre di una ROM per la programmazione avanzata, con istruzioni in BASIC che migliorano le capacità di gestione dei dati, l'ordinamento e la selezione di matrici, le trasformazioni dei caratteri minuscoli e maiuscoli.
La tastiera e il set di caratteri standard sono l'USASCII.

HEWLETT - PACKARD - MILANO

Il Compucolor II in Italia

Il Compucolor II è un personal computer progettato e costruito dalla Compucolor Corporation di Norcross (Georgia), azienda collegata a industrie con lunga esperienza di produzione di terminali intelligenti a colori e grafici. La Compucolor ha inteso pertanto realizzare un calcolatore completo, con schermo a colori di 13 pollici e ad un costo inferiore a quello di un personal computer in bianco e nero.
Il Compucolor II ha capacità grafiche utilissime nell'elaborazione di dati gestionali e scientifici; permette la programmazione di otto colori (nero, rosso, verde, giallo, blu, magenta, viola, bianco), per evidenziare dati in forma grafica e alfanumerica; è dotato, nella versione standard, di un mini-floppy-disk della capacità di 51,2 + 51,2 Kbytes, più che sufficiente per la maggior parte delle applicazioni; ha un I/O RS-232C (con velocità di trasmissione selezionabile da software) che gli consente di diventare un terminale intelligente e di collegarsi ad una vasta serie di periferiche.
La tastiera fornita nella versione standard è di tipo professionale, con la possibilità di avere in opzione altre versioni che rappresentano

quanto di più completo si possa desiderare.
Il microprocessore utilizzato è l'8080A; la memoria dell'unità base è di 16 Kbytes, ampliabile a 32 Kbytes; il linguaggio di programmazione è il BASIC 8001. Le periferiche fornite dalla Compucolor come opzione sono costituite da più unità a mini-floppy-disk aggiuntive da 51,2 + 51,2 Kbytes.

SYMIC - MILANO

CORSI

Corso di livello 1 sulla applicazione dei sistemi di elaborazione della General Processor

Nei giorni 9, 10, 11 Ottobre si terrà a Firenze, presso l'Hotel Michelangiolo il corso applicativo di primo livello, dedicato cioé a persone tecnicamente non troppo preparate, sulle applicazioni di carattere commerciale del personal computer, con particolare riferimento ai sistemi della General Processor.
Alle conferenze parteciperanno anche commercialisti e fiscalisti, nell'intento di fornire una visione più direttamente applicativa del problema. Tra gli argomenti trattati saranno la fatturazione, l'emissione di vari documenti tra cui le bollette di accompagnamento, la gestione degli archivi ecc. Le lezioni saranno integrate da eserci-

tazioni pratiche. La partecipazione al corso costa 50.000 lire più IVA a puro titolo di rimborso spese. Le prenotazioni possono essere fatte via telefono nelle ore del mattino al 21.91.43 di Firenze (prefisso 055).

GENERAL PROCESSOR - FIRENZE

Corsi Syntax sui microprocessori Fairchild

La Fairchild Semiconduttori S.p.A. per il lancio sul mercato italiano del suo microprocessore 9440 MICROFLAME ha predisposto un esteso calendario di corsi, affidandone la completa realizzazione agli specialisti del "Settore Formazione" della Syntax S.p.A., che curerà anche tutti gli aspetti logistici dell'operazione, realizzando i corsi presso la sua sede di via Vincenzo Monti, 47 a Milano.

Com'è noto, il 9440 MICROFLAME è un microprocessore a 16 bits, che esegue un set di istruzioni multifunzione, con potenze di elaborazione da minicomputer, è contenuto in un singolo chip.

I corsi sul 9440 MICROFLAME avranno la durata di tre giorni e si svolgeranno secondo il seguente programma: Introduzione sui microprocessori e sul sistema - Presentazione dei componenti 9440, 8441, 9442, Set di istruzioni - Spark 16 e Firebug - Esercitazioni sullo Spark 16 - Altri prodotti software e hardware.

Le prime 6 edizioni del corso saranno realizzate dalla Syntax a partire dai giorni: 10 settembre, 1 ottobre, 22 ottobre, 12 novembre, 3 dicembre, 7 gennaio 1980.

Con questa iniziativa la Syntax, una delle maggiori società europee di consulenza e produzione software, accentua la sua presenza in un settore del mercato, quello dei "micro", che si va sempre più imponendo all'attenzione dei tecnici e degli utenti, sia per i tassi di espansione che lo caratterizzano, sia per le notevoli possibilità applicative consentite dall'impiego dei microprocessori.

SYNTAX - MILANO

"Continuing Education" Programma 1980

È in preparazione il programma dei corsi e seminari che la MICROLEM divisione didattica organizzerà nel 1980. Il metodo sperimentale "hands-on", già usato con successo nei precedenti corsi, verrà utilizzato ampiamente anche nel 1980, mentre nuovi sistemi didattici verranno affiancati all'MMD1.
Il programma prevede corsi di durata variabile sui diversi temi dell'informatica, dall'elettronica digitale ai microcomputer, con particolare riguardo alle tecniche di programmazione e di interfacciamento.
8080A, 8085, Z-80 e 6800 sono i microprocessori che saranno oggetto di studio per il programma del 1980. Temi di particolare attualità, quali l'utilizzazione dei micro nei controlli industriali, la ricerca e la riparazione dei guasti nei sistemi di controllo a microprocessore, l'introduzione ai microcomputer per i dirigenti d'azienda, saranno trattati in modo chiaro ed esauriente da specialisti della materia, nello spirito della "Continuing Education".
Per la prima volta faranno la loro comparsa docenti italiani, istruiti per questi tipi di corsi sperimentali. Per il 1980 la MICROLEM divisione didattica organizza anche **corsi "in-house"** per le aziende che abbiano particolari esigenze di preparazione del personale: i programmi specifici per questo tipo di corsi vengono definiti di volta in volta, d'accordo col cliente.
Per tutti i corsi la MICROLEM fornisce in uso le attrezzature necessarie all'esecuzione degli esperimenti. A tutti i partecipanti vengono rilasciati i libri di testo e la letteratura tecnica eventualmente utilizzata dai docenti. Il costo medio dei corsi brevi, generalmente della durata di 5 giorni, oscilla fra le 80 e le 100 mila lire al giorno per persona. Il numero massimo dei partecipanti ammessi ai corsi è 30. Ad ogni corso viene normalmente garantito un adeguato servizio di traduzione simultanea.
Nel 1980 i corsi potranno avere luogo nei nuovi locali, attualmente in corso di allestimento, appositamente attrezzati dalla MICROLEM divisione didattica in via Vittor Pisani 22, Milano.
Il programma dettagliato dei Corsi e Seminari 1980 sarà disponibile nel mese di novembre.

MICROLEM - MILANO

INFORMATICA

Un premio europeo per la ricerca in informatica

La CII Honeywell Bull di Parigi, consociata a maggioranza francese del gruppo Honeywell Information Systems, ha istituito un premio intitolato "Premio Europeo CII Honeywell Bull per la Ricerca". Il premio verrà assegnato una volta ogni due anni a uno o più ricercatori dei paesi dell'Europa Occidentale che abbiano recato, nel corso del periodo, un contributo significativo allo sviluppo dell'elaborazione dati. Secondo il regolamento del Premio, il lavoro di ricerca deve essere stato condotto interamente o prevalentemente in Europa. Esso inoltre deve essere rivolto ad applicazioni pratiche, anche se inizialmente limitate a un'area sperimentale.

Il premio CII Honeywell Bull, che avrà un valore di 50 mila franchi francesi, verrà assegnato per la prima volta a Parigi nel dicembre di quest'anno. La scelta dei candidati e la proclamazione del vincitore avverranno ad opera di un comitato scientifico di dodici esperti, provenienti dal mondo industriale e scientifico, presieduto dal direttore del Centro di Ricerca della CII Honeywell Bull, François Maison.

La consegna del premio verrà fatta da M. Pierre Agrain, Segretario di Stato francese per la Ricerca.

Le domande di ammissione e la documentazione dei lavori dovranno pervenire (per la prima edizione del premio) alla CII Honeywell Bull entro il 1° ottobre 1979. I relativi moduli sono già stati distribuiti alle Università e alle organizzazioni di ricerca dei vari paesi.

Chi fosse interessato può comunque richiederli a:

M. François Maison
"European Prize for Research"
CII Honeywell Bull
68, route de Versailles
78430 Louveciennes (Francia)

HONEYWELL INFORMATION SYSTEM ITALIA - MILANO

Comunicato n.1 ai Rivenditori di Macchine per Ufficio.

Cromemco Sistema Tre: o lo vendete voi, o lo vendono gli altri.

Cromemco Sistema Tre: il computer professionale dal prezzo abbordabile.

Il Sistema Tre della Cromemco è al vertice nel settore dei personal computers. Sia per le sue caratteristiche, sia per le possibilità che offre in ogni tipo di attività: gestione aziendale, trattamento della parola, gestione di banche di dati. Ma anche calcolo scientifico, controlli industriali, e così via.

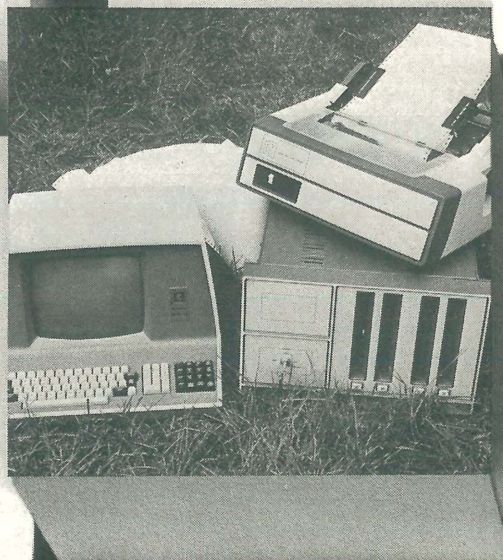
Il Sistema Tre della Cromemco è un sistema basato su microprocessore Z 80: espandibilità fino a 512 KByte di memoria centrale, 2 o 4 drive per floppy-disk per un totale di 1 MByte di memoria su disco, oppure unità hard-disk da 11 MByte o 22 MByte, collegamento a unità video con tastiera e stampanti.

Il vasto corredo di software include un sistema operativo a dischi e i linguaggi COBOL, Extended BASIC, Multiuser BASIC (fino a sette utenti in time-sharing), FORTRAN IV. Sono inoltre offerti un Data Base Management System ed un Word Processing.

Con la Unicomp cominciate a vendere subito. E tutto quel che vi serve è un limitato investimento.

La Unicomp vi permette di limitare al massimo gli investimenti necessari per cominciare a vendere sistemi Cromemco e altri personal computers.

Perché, oltre alle macchine, noi vi diamo una gamma di servizi che non ha l'uguale in questo settore.

- Tutto l'addestramento che serve ai vostri tecnici, per metterli in grado di identificare eventuali guasti a livello di scheda.
- Un laboratorio riccamente attrezzato e con personale altamente qualificato per riparare, se lo desiderate, le schede difettose o eventuali unità periferiche.
- Parti di ricambio sempre disponibili e una documentazione sempre aggiornata, per permettervi di offrire ai vostri clienti un servizio di assistenza professionale ed efficiente.
- Addestramento dei vostri programmatori sul software di base.
- Raccolta di programmi standard: contabilità generale, contabilità IVA, clienti e fornitori, gestione di magazzino, fatturazione, etc. La disponibilità di questi programmi, cosa molto importante per ridurre i vostri investimenti nello sviluppo di procedure applicative, vi mette subito in grado di eseguire dimostrazioni e di accettare ordini dai vostri clienti. E su questi programmi, concepiti per essere uno strumento nelle vostre mani, potete intervenire facilmente per modificarli e renderli "su misura" delle specifiche esigenze dei vostri clienti.

UNICOMP

i piccoli sistemi di grande avvenire.

20092 cinisello balsamo (milano)
palazzo testi/via cantù, 20
tel.(02) 6121041 (5 linee)

è marchio registrato della Unicomp S.r.l.

Introducing the personal computer you've waited for. The Exidy Sorcerer.

I didn't buy my personal computer until I found the one that had all the features I was looking for.

The Exidy Sorcerer does everything I wanted to do and a few things I never dreamed of.

It isn't magic. Exidy started with the best features of other computers, added some tricks of their own, and put it all together with more flexibility than ever before available. Presto! My reasons for waiting just disappeared.

I wanted pre-packaged programs. Software on inexpensive cassette tapes for the Sorcerer is available from Exidy and many other software makers.

I wanted user programmability. The Sorcerer's unique plug-in ROM PAC™ Cartridges contain programming languages such as Standard (Altair 8k*) BASIC, Assembler and Editor (so I can develop system software), operating systems such as DOS (so I can also use FORTRAN and COBOL) and applications packages such as Word Processor.

I wanted graphics, and the Sorcerer is super. Its 256 character set—more than any other personal computer—includes 128 graphic symbols that I can define.

I wanted high resolution video. With 122,880 points in a 512 x 240 format, I get the most detailed illustrations.

I wanted to display more information. The Sorcerer displays 1920 characters in 30 lines of 64 characters —equal to a double-spaced typed page.

I wanted a full, professional keyboard. The Sorcerer's 79-key data processing keyboard provides designated graphics, the complete ASCII character set in upper and lower case, and a 16-key numeric pad.

I wanted memory. The 12k of ROM holds a Power-On Monitor and Standard BASIC; the 8k of RAM is internally expandable to 32k.

I wanted expandability. Serial and parallel I/Os are built in, and the optional 6-slot S-100 expansion unit lets my system grow.

I wanted a computer that's easy enough for children to use. I just connect my Sorcerer to a video display and a cassette tape recorder, and if I have any questions the easy-to-understand Operation and BASIC Porgramming manuals have the answers.

I wanted to buy from an experienced manufacturer. In five years Exidy has become the third largest producer of microprocessor-based video arcade games.

I wanted to spend less than a thousand bucks. (This is where Exidy does a little magic.)

* Altair is a trademark of Pertec Computer Corp.

Distribuito da:

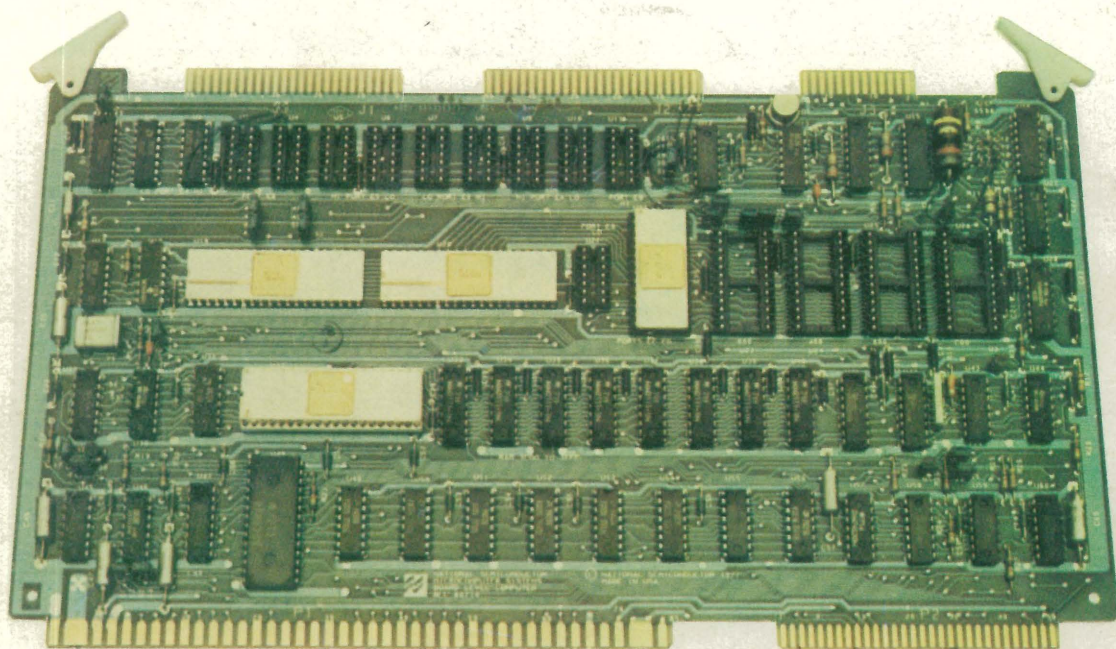
REDIST

Divisione della

G.B.C. italiana

V.le Matteotti, 66 - 20092 Cinisello B. (MI)
Tel. (02) 6181801 - 6189391 - Telex: 330028

Serie 80 BLC
componenti per microcomputers

La National Semiconductor per un Sistema Globale offre sottoassemblaggi per la realizzazione di sistemi completi in ogni gamma di applicazione.
La National, al fine di permettere agli utilizzatori un rapido servizio per una progettazione ottimizzata in tempo, caratteristiche, costi, mette a disposizione oltre ai circuiti stampati già pronti e collaudati, anche i supporti necessari per lo sviluppo del hardware e software, nonché gli strumenti per la simulazione.

TIPO	DESCRIZIONE	TIPO	DESCRIZIONE
BLC-016	16K Byte RAM Board	BLC-80/11	S/80 Computer W/8KB EPROM Capability
BLC-032	32K Byte RAM Board	BLC-80/12	BLC-80/11 W/2K Bytes RAM
BLC-048	48K Byte RAM Board	BLC-80/14	BLC-80/11 W/4K Bytes RAM
BLC-064	64K Byte RAM Board	RMC-80/14	Rack Mount Computer - BLC-80/14 Based
BLC-104	4K Byte RAM and 8K ROM and I/O Board	BLC-80/204	Board Level Computer
BLC-116	16K Byte RAM and 8K ROM and I/O Board	RMC-80/204	Rack Mount Computer System
BLC-416	16K Byte ROM/PROM Expansion Board	BLC-80P	Series 80 Prototyping Package
BLC-501	Direct Memory Access Board	BLC-80P	Prototyping Package
BLC-508	I/O Board 4 Input + 4 Output (8-Bit) Ports	BLC-80P14	Proto Package Based on BLC-80/14
BLC-517	Expansion I/O Board 48 Lines & RS232C	BLC-80P204	Prototyping Package
BLC-530	RS232C to TTY Adapter	BLC-8432	32K Byte ROM/PROM Expansion Board
BLC-8534	4 Channel Programmable Serial I/O Board	BLC-8538	8-Channel Programmable Serial I/O Board
BLC-556	Optically Isolated I/O Board	BLC-8610	Extension Board with Power Control
BLC-604	Card Cage 4 Slot	BLC-8905	Prototype Board
BLC-610	Extender Board	BLC-901	Terminator Module 220/330 Ω
BLC-614	Expansion Card Cage, 4 Slot	BLC-902	Terminator Module 1K Ω
BLC-635	Power Supply	BLC-903	Option Jumper Kit
RMC-660	Rack Mount System W/8 Slots + Hd Power Supply	BLC-905	Prototype Board
BLC-665	Heavy Duty Power Supply	BLC-910	80/10 System Monitor Firmware
BLC-711	Analog Input Board	BLC-920	80/204 System Monitor Firmware
BLC-724	Analog Output Board	BLC-955	Serial I/O Cable Kit
BLC-732	Combo Analog I/O Board	BLC-956	Parallel I/O Cable Kit
BLC-80/10	Series 80 Microcomputer	BLC-957	Power Supply Cable Kit
RMC-80/10	Series 80 Rack Mount Computer		

National Semiconductor, Via Solferino, 19
20121 Milano (02) 661542 - 630410 - Tx: 332835

AGENTE: **Repco srl,** Milano (02) 4985274-4985932-4985494, Roma (06) 8107788
DISTRIBUTORI: **Adelsy spa, Milano** (02) 4985051, Genova (010) 589674,
Udine (0432) 26996, Padova (049) 45600-45778, Torino (011) 539141, Roma (06) 594559
• **E.D.L. spa,** Napoli (081) 632335-611988 • **Esco Italiana,** Milano (02) 6072441-5
• **Intelco,** Bologna (051) 726186, Firenze (055) 608107 • **Inter-Rep spa,** Torino (011) 752075/6/7
• **Intesi,** Milano - S. Donato Milanese (02) 51741, Roma (06) 2275130-223372,
Torino (011) 613963 • **Side srl,** Ancona - Osimo Scalo (071) 79307-79017
DISTRIBUTORE SISTEMI DI MEMORIE: **ESE srl,** Milano (02) 600733/973-6882334